

上海迅时通信设备有限公司

OM API 技术开发接口 (V7.0)

网址: www.newrocktech.com.cn

电话: 021-61202700

传真: 021-61202704

文档版本: 2021-4



文档版本 02 (2021-04)

技术内容更新

文档版本 01 (2020-12)

全文

1. 快速入门	6
1.1 初次见面	6
1.1.1 关于 OM API	6
1.1.2 OM API 有如下四类本领	6
1.2 交互原理	6
1.2.1 传输协议	7
1.2.2 通信方式	8
1.3 测试 demo	9
1.3.1 配置	9
1.3.2 跑个流程	11
1.4 API 认证	14
1.4.1 正向认证	14
1.4.2 反向认证	16
2. OM API 接口	18
2.1 控制命令(Control)	18
2.1.1 查询(Query)	18
2.1.2 配置(Assign)	33
2.1.3 呼叫保持和接回(Hold/Unhold)	40
2.1.4 静音开启与解除(Mute/Unmute)	41
2.1.5 监听和插播(Monitor/Talk/Listen)	43
2.1.6 强插(Bargein)	47
2.1.7 强拆(Clear)	49
2.2 呼叫转接命令(Transfer)	50
2.2.1 连接(Connect)	50
2.2.2 队列(Queue)	65
2.2.3 会议(Conference)	67
2.3 语音管理命令(Manage)	69
2.3.1 查询语音文件	69
2.3.2 删除语音文件	70
2.4 来电受理(Accept)	71
2.5 事件报告	72
2.5.1 系统事件	72
2.5.2 分机状态变更事件	73
2.5.3 呼叫状态变更事件	74
2.5.4 来电呼入控制流程事件	80
2.5.5 按键信息事件(DTMF)	82
2.5.6 语音文件播放完毕事件(EndOfAnn)	84
2.5.7 分机组队列事件(Queue)	85
2.6 通话记录报告	87
2.6.1 呼入(IN)	90
2.6.2 呼出(OU)	92
2.6.3 内部互拨(LO)	93
2.6.4 呼叫转移(FI/FW)	94

2.6.5 双向外呼(CB)	96
3. 实现方案	98
3.1 点击拨号	98
3.1.1 实现原理	98
3.1.2 步骤描述	98
3.1.3 OM API 点击拨号的特点	99
3.2 来/去电弹屏	99
3.2.1 实现原理	100
3.2.2 来电弹屏	100
3.2.3 去电弹屏	101
3.2.4 B/S 架构下如何实现来去电弹屏	101
3.3 多级 IVR	102
3.3.1 配置要求	102
3.3.2 相关 API 列举	102
3.3.3 简易的 IVR 实现流程图	103
3.3.4 异常处理	103
3.4 来电黑名单	106
3.5 来电直通车	108
3.5.1 痛点	108
3.5.2 简介	108
3.5.3 配置要求	108
3.5.4 实现流程	108
3.6 指定中继外呼	109
3.6.1 加前缀外呼	109
3.6.2 指定 trunk id 外呼	110
3.7 酒店叫醒解决方案	111
3.7.1 场景描述	111
3.7.2 实现方案	112
3.8 API+TTS 实现方案	114
3.9 语音播报实现方案	118
3.10 API 中间件解决方案	120
3.10.1 OM 配置	121
3.10.2 API 中间件配置	121
3.10.3 应用服务器配置	124
3.11 API+依好解决方案	125
3.12 分机外呼控制实现方案	128
3.13 监听+插播+强拆实现方案	130
3.14 接通后播放语音实现方案	132
3.15 满意度评价实现方案	134
3.16 公网服务器访问私网内 OM	137
3.16.1 端口映射	138
3.16.2 客户端直接访问 OM	142
4. 专题介绍	145
4.1 IPPBX	145

4.2 外线和分机	146
4.2.1 分机	146
4.2.2 外线	146
4.3 语法和对象	147
4.4 API 认证	148
4.4.1 IP 认证	148
4.4.2 数字签名认证	149
4.5 API 反向认证	153
4.6 API 功能开关	155
4.7 API 控制模式功能介绍	158
4.8 外转外功能开关	159
4.9 录音	160
4.10 语音文件配置说明	165
4.10.1 Web 上传	165
4.10.2 远程下载	165
4.11 语音菜单	166
4.11.1 基本原理	166
4.11.2 三大特点	166
4.11.3 应用场景	167
4.11.4 配置	167
4.11.5 语音文件的播放说明	168
4.11.6 语音菜单相关接口	169
4.12 出队列判断方法	170
4.13 插播 TTS 远程语音文件	172
4.14 跨域访问 OM	173
4.14.1 简介	173
4.14.2 限制条件	173
4.14.3 解决方法	173
4.15 Telnet 后台访问 OM	176
5. 常见问题解答	180
5.1 网络环境相关问题解答	180
5.2 API 认证相关问题解答	180
5.3 menu 相关问题解答	181
5.4 配置相关问题解答	181
5.5 呼叫相关问题解答	182
5.6 事件相关问题解答	183
5.7 CDR 相关问题解答	185
5.8 其他问题	186
6. 附录	187
6.1 系统参数列表	187
6.2 系统参数查询和配置方法	190
6.2.1 查询方法	190
6.2.2 配置方法	191
6.2.3 响应结果	192

1 快速入门

本章简单介绍：OM API 基本概念及应用服务器和 OM 之间的基本交互原理，手把手教你完成 OM API 认证配置，并使用测试工具体验应用服务器和 OM 之间的消息交互，助您快速入门。

1.1 初次见面

1.1.1 关于 OM API

迅时 OM 系列的 IPPBX 设备提供统一的开放式接口——OM API，开发者可以通过 OM API 对 OM 设备进行控制、监控和数据统计等。

OM API 本质上是经过封装的简单 XML 消息，应用服务器和 OM 设备之间通过 HTTP 协议进行通信。

- 1) **应用服务器通过 OM API 对 OM 进行操作：**参数查询、参数配置、状态查询、呼叫控制、呼叫转接等。
- 2) **OM 实时向应用服务器推送报告：**分机状态变化、呼叫状态、配置变化、服务启动、DTMF、语音播放完毕和通话记录等。

OM API 使 OM 设备具有更大的灵活性和可操作性，开发者可利用 OM API 开发或对接呼叫中心、计费系统、酒管系统、CRM、OA 办公系统等丰富多彩的应用。

1.1.2 OM API 有如下四类本领：



参数和状态查询



参数配置



呼叫操控



事件监控

可实现的主要功能有：点击拨号、来/去电弹屏、通话记录、录音、状态监控、分机组和队列、IVR 语音导航、满意度调查、酒店叫醒服务、语音信箱、语音验证码、来电黑/白名单。

1.2 交互原理

简单认识 OM API 之后，我们来了解下 OM API 的通信方式。

1.2.1 传输协议

应用服务器和OM之间基于HTTP协议进行通信，API消息封装在HTTP包体中。



消息内容如：

HTTP 请求消息：

```
POST /xml HTTP/1.0
Content-Type:text/xml
Content-Length:101

<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Query">
  <DeviceInfo/>
</Control>
```

HTTP 响应消息：

```
HTTP/1.0 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<DeviceInfo>
  <manufacturer>New Rock Technologies, Inc</manufacturer>
  <model>Rev 1.0.1 WROC2000-1S/1</model>
  <version>Rev 2.2.5.81.1</version>
  <mac>00:0E:A9:00:12:BD </mac>
  <devices>
    <ext lineid="Phone 1" id="200" />
    <ext lineid="IPPhone 50" id="208" />
    <line lineid="Line 2" id="02161208234" />
    <line lineid="IPLine 21" id="02161204000" />
  </devices>
</DeviceInfo>
```

1.2.2 通信方式

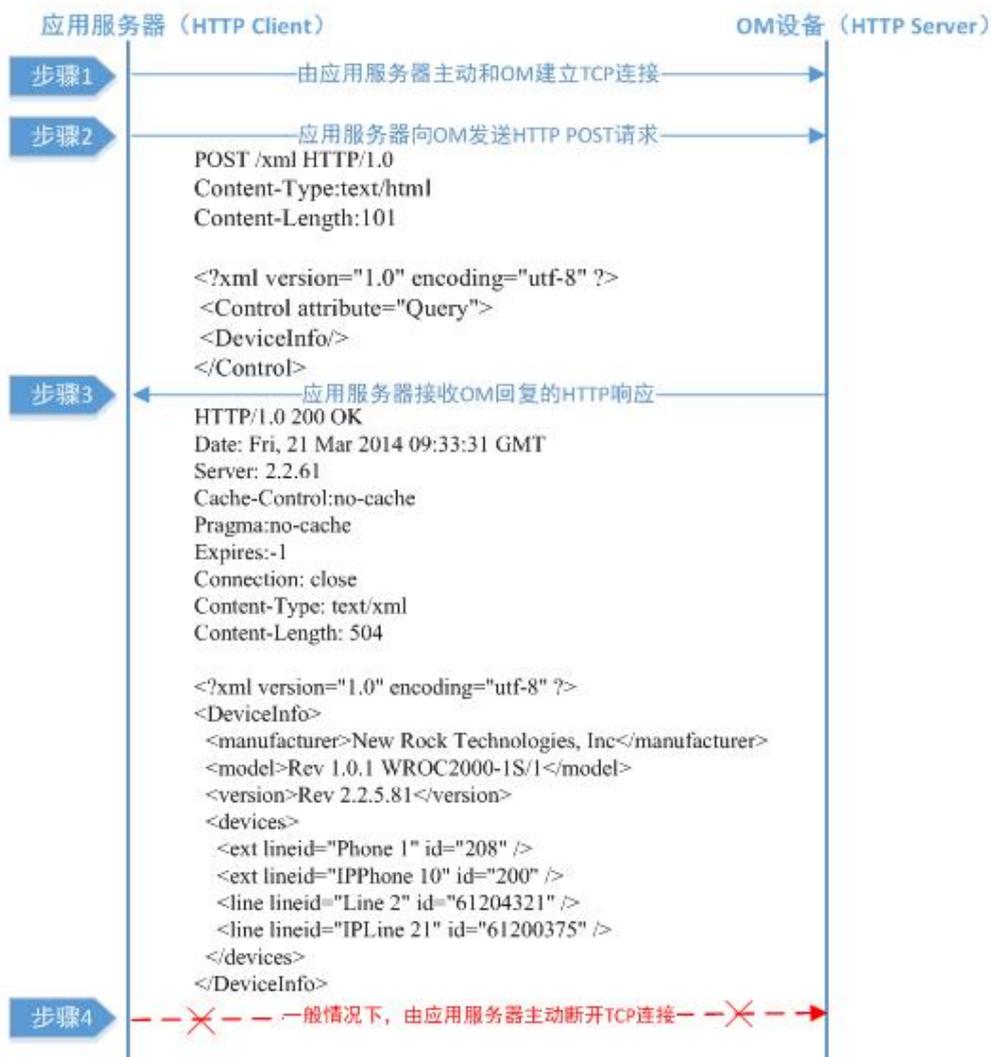
应用服务器和 OM 之间的交互是双向的，双方互为 HTTP 服务端和客户端。

正向：应用服务器作为 HTTP 客户端，OM 作为 HTTP 服务端。

应用服务器请求 OM 执行某个功能（如，发起呼叫）或提供某些信息（如，查询状态）。此时，采用的是 HTTP POST 方法、TCP 短连接方式（注：OM 只支持 TCP 短连接接收）。

流程为：①请求 ②响应 ③断开。

交互图如下：



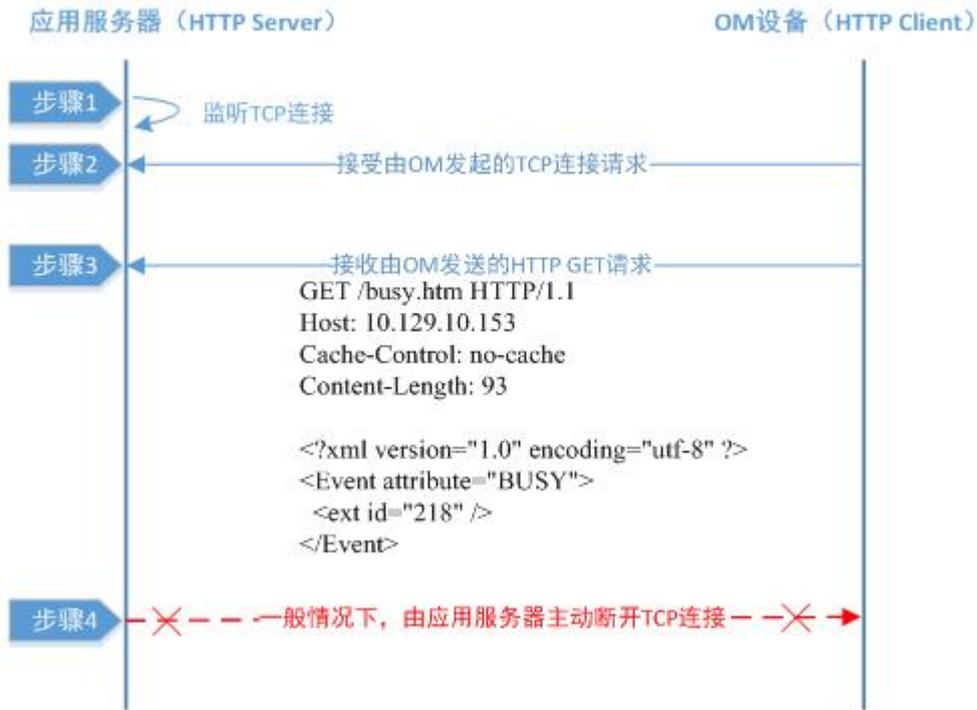
反向：OM 作为 HTTP 客户端，应用服务器作为 HTTP 服务端。

OM 主动向应用服务器推送某些消息（如，分机振铃事件），应用服务器收到消息后断开 TCP 连接（注：这里不是标准的 HTTP 请求响应流程，不需要应用服务器回复响应）。

此时，采用的是 HTTP GET 或 POST 方法（默认为 GET，若参数 `API_METHOD = 1`，则为 POST）。

注：OM 默认以短连接方式推送消息，也可通过参数 `CONTROL_TYPE` 配置为长连接。建议采用短连接方式。

流程为：①接收消息 ②断开。



1.3 测试 demo

我们用测试工具（相当于一个简单的应用服务器）来演示 OM 和 API 应用服务器的配置方法及收发消息过程。

1.3.1 配置

注：这里我们采用 IP 认证方式。

步骤 1：配置 OM 设备认证地址。

登录 OM 设备页面，点击应用服务器 > API，在应用服务器一栏选择自定义（默认选中为依好，依好是内置在设备里的小型呼叫中心），填写应用服务器地址（即测试工具所在的电脑 IP：OM 发送端口，可自定义）如，192.168.130.27:8989。

步骤 2：配置分机和外线的 API 开关。

进入应用服务器 > API，在 API 功能开关一栏将分机和外线的状态监控、来电应答前/来电应答后控制开关打开，点击保存，并重启设备。

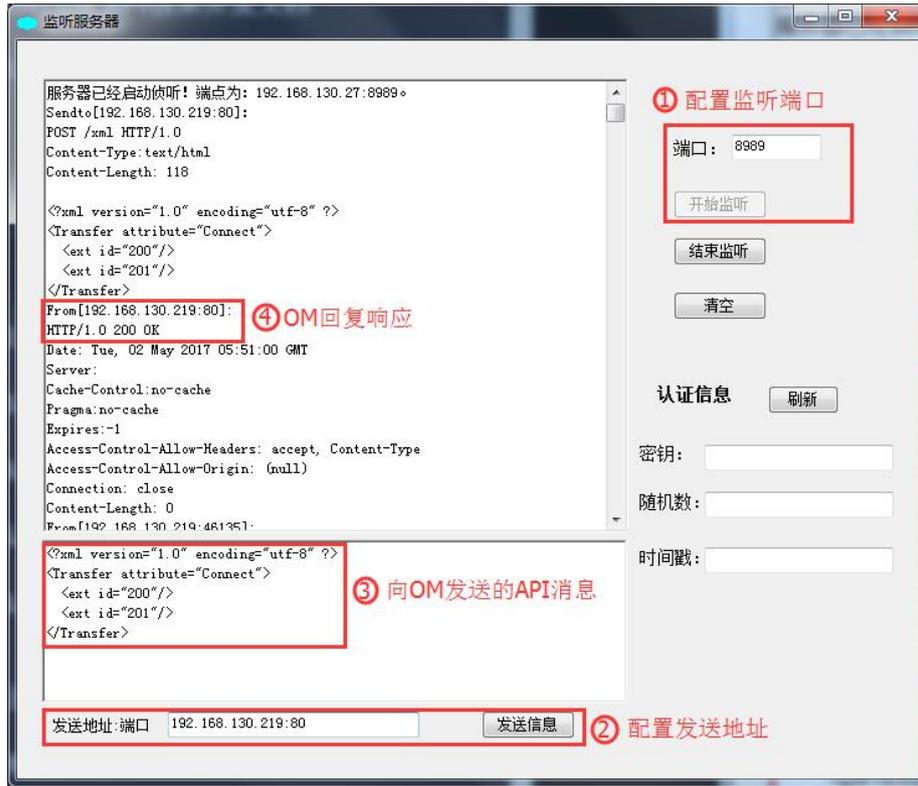
设备配置如下图所示：

**步骤 3：配置测试工具。**

- 1) 点击[这里](#)下载测试工具及其源代码，并在电脑桌面打开，填写一个 HTTP 监听端口（该端口与设备上配置的 OM 发送端口保持一致，应为 8989），点击开始监听。（注：测试工具的源码可[点击这里](#)下载）。
- 2) 填写发送地址（即 OM 的 IP 地址和 HTTP 端口），如 192.168.130.219:80。

注：OM 的 HTTP 端口默认为 80，可在设备高级设置>安全配置>Web 管理处修改。

测试工具配置如下图所示：



1.3.2 跑个流程

完成了配置以后，接下来我们演示一个分机呼叫分机的流程，为第二章的接口使用和理解奠定基础。

发送命令：

如何通过 OM API 实现分机呼分机呢？

只需向 OM 发送一条 API 消息：

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <ext id="200"/>
  <ext id="201"/>
</Transfer>
```

说明：

- 1) 第一行是 XML 声明，每个 API 消息都有且相同。它定义了 XML 的版本（1.0）和所使用的编码方式（utf-8）。

- 2) 第二行是 XML 的根。根元素 **Transfer** 表明这个是一个呼叫转接类的 API。属性值 **Connect** 表示本次转接的属性为连接。
- 3) 第三、第四行中 **ext** 是“分机”的英文单词 **extension** 的简写，200 为主叫分机号码，201 为被叫分机号码。
- 4) 第五行为根节点的闭合标签。

注：更多 XML 的语法参见 XML 教程，更多 OM API 语法详解请参见第二章。

观察执行结果：

执行完该 API 后，后续流程为：

- 1) 主叫分机 200 会先振铃(默认，先呼谁后呼谁由参数 **API_CALLING** 控制)；
- 2) 将 200 摘机后，被叫分机 201 开始振铃，并且 200 可以听到“嘟嘟”的回铃音；
- 3) 将 201 也摘机后，双方成功建立通话；
- 4) 任意一方挂机后通话结束。

观察收到的 API 消息：

查看测试工具，可以看到接收到很多条 API 消息。这些消息中，有两个消息是话单 (CDR)，其他的为事件 (Event)。

- 1) CDR 为通话记录，在通话结束时产生。2 个 CDR 中一个是主叫分机的通话记录，另一个是被叫分机的通话记录；
- 2) Event 表示事件消息，由呼叫过程中 OM 自动触发。

事件：

本次呼叫过程中，收到的事件属性有这些：BUSY、IDLE、RING、ALERT、ANSWER、ANSWERED、BYE。

其中：

- 1) BUSY 和 IDLE 是一对，在分机状态发生变化时产生。BUSY 表示分机由空闲变为忙状态，IDLE 表示分机由忙变为空闲状态；
- 2) RING、ALERT、ANSWER、ANSWERED、BYE 属于一个系列，在呼叫过程中产生。
其中：

A.RING 和 ALERT 是一对，RING 表示分机开始振铃，ALERT 表示收到对方的回铃（ringback）信号。

B.ANSWER 和 ANSWERED 是一对，ANSWER 表示分机应答，ANSWERED 表示收到对方应答的信号。

C.BYE 表示通话结束。

通过以上事件，你可以实时监控分机的线路状态和呼叫情况，并可以实现一些应用功能，比如：来/去电弹屏（当分机振铃时将来电号码对应的客户资料弹屏显示在电脑屏幕上）。

注：更多关于事件的介绍，参见 2.5 章节。

通话记录（CDR）：

通话结束后，OM 会立即将通话记录推送给应用服务器（这里指测试工具）。

消息格式：

```
<?xml version="1.0" encoding="utf-8" ?>
<Cdr id="13620170308103713-0">
  <callid>32820</callid>
  <TimeStart>20170308103709</TimeStart>
  <Type>IN</Type>
  <Route>IC</Route>
  <CPN>200</CPN>
  <CDPN>201</CDPN>
  <TimeEnd>20170308103713</TimeEnd>
  <Duration>2</Duration>
  <TrunkNumber></TrunkNumber>
  <Recording>20170308/200_201_20170308_103711_8034_cd.wav</Recording>
  <RecCodec>PCMU</RecCodec>
</Cdr>
```

参数说明：

参数名称	解释说明
Cdr id	通话记录的编号。格式：系列号+年月日时分秒+固定内容(-0)
callid	通话的相对唯一标识符

参数名称	解释说明
TimeStart	呼叫起始时间戳，格式：年月日时分秒
Type	话务类型，IN 表示呼入，LO 表示内部呼叫
Route	路由类型，IC 表示内部路由
CPN	主叫号码
CDPN	被叫号码
TimeEnd	呼叫释放时间戳，格式：年月日时分秒
Duration	通话时长，单位：秒。即，从呼叫接通到呼叫释放的时长，不包括振铃时间。
Trunk	中继号码（本次是内部呼叫，没有用到中继，所以值为空）
Recording	录音文件的相对保存路径，格式：生成日期/录音文件名称

注：更多关于 CDR 的介绍，参见 [2.6 章节](#)。

1.4 API 认证

API 认证方式包括正向认证和反向认证两种。

正向认证：应用服务器向 OM 发送请求命令时，需通过 OM 的认证。

反向认证：OM 向应用服务器发送消息时，需通过应用服务器的认证。

1.4.1 正向认证

正向认证分为 IP 认证和数字签名认证两种方式，您可以根据自己的实际场景选择一种来完成认证。

IP 认证：

IP 认证，只允许某一固定 IP 地址向 OM 发送 API 请求，其他地址统统认为没有权限。

适用的应用场景：

- 1) 应用服务器的 IP 地址/域名固定；
- 2) 一个应用服务器对接一台 OM；
- 3) 应用服务器和 OM 之间网络互通。

配置:

配置方法，如下图所示：



参数说明：

- 1) **服务器地址：**应用服务器的 IP 地址/域名和监听端口，如：192.168.130.27:8989。
如果用户未指定端口时默认为 80 端口。
- 2) **URL：**接收 API 报告的相对路径（也可不填写）。格式为：
{part1}/{part2}/{part3}/{.....}，如：omapi/report。

服务器地址和 URL 组合起来即为应用服务器接收 API 报告的全路径，如：
192.168.130.27:8989/omapi/report。

应用服务器地址的作用：

- 1) **接收 API 报告：**OM 会将 API 消息推送给这个地址；
- 2) **访问权限控制：**OM 只受理从该服务器的 IP 地址（端口不影响）发送的 API 请求；
拒绝受理从其它地址发送的 API 请求，并对该请求响应 Unauthorized。

点击[这里](#)查看更多详情。

数字签名认证:

数字签名认证本质上是通过验证 OM 和应用服务器双方持有的秘钥来完成认证。

版本要求:

OM 软件版本：Rev 2.1.5.99 及其以上。

适用的应用场景:

- 1) 应用服务器采用动态域名, IP 地址不固定;
- 2) 一个 OM 要对接多个 API 应用服务器;
- 3) API 应用客户端要直接访问 OM;
- 4) API 消息的源地址容易发生变化;

配置:

配置参数: 数字签名认证密码 (接收) 和数字签名有效期。

配置界面, 如下图所示:



参数说明:

- 1) **API 数字认证密码 (接收):** OM 对应用服务器认证的密钥, 可自定义, 需和应用服务器发送请求时携带的密钥保持一致。
- 2) **API 数字认证有效期:** 可自定义, 范围: 0~86400, 单位: 秒, 只有在该有效期内认证参数才有效, 0 表示永久有效。

点击[这里](#)查看更多详情。

注: 不论是 IP 认证还是数字签名认证, 都需要在 OM 页面配置应用服务器地址, 用来接收 API 消息。

1.4.2 反向认证

反向认证, 即应用服务器对 OM 的认证。OM 向应用服务器推送 API 消息时携带 Auth 认证信息, 应用服务器根据收到的消息是否满足认证条件来选择是否接收该消息。

注：只有应用服务器要求对 OM 进行认证时才配置，若不要求，可不配置。

版本要求：

Rev 2.1.5.116 及以上

配置：

配置参数：数字签名认证密码（发送）和数字签名有效期。

配置界面，如下图所示：



参数说明：

- 1) **API 数字认证密码（发送）**：应用服务器对 OM 进行认证的密钥，可自定义，需和应用服务器接收消息时配置的密钥保持一致。
- 2) **API 数字认证有效期**：可自定义，范围：0~86400，单位：秒，只有在该有效期内认证参数才有效，0 表示永久有效。

[点击这里查看更多详情。](#)

2 OM API 接口

本章为 OM API 接口部分，包括 **API 请求命令**和 **API 报告**两部分内容。

- 1) API 请求命令，指应用服务器向 OM 发送的 API 消息，包括**制命令**、**转接命令**和**来电受理命令**三种类型。
- 2) API 报告，指 OM 主动向应用服务器推送的 API 消息，包括**事件报告**和**通话记录报告**两种类型。

2.1 控制命令(Control)

控制命令包括查询、配置、呼叫保持与接回、静音与解除静音、监听、强插、强拆。

2.1.1 查询(Query)

此类 API 用于查询 OM 设备上指定对象的相关信息(如，配置参数和状态)。这些对象包括：设备信息(deviceInfo)、分机(ext)、中继(trunk)、来电(visitor)、去电(outer)、分机组(group)、语音菜单(menu)。

查询请求的规则说明：

最小的查询单位是对象，即不支持单独查询该对象的某一个具体参数。

查询结果的规则说明：

- 1) 查询结果中包含该对象的所有可提供的相关参数和状态信息。
- 2) 如果查询结果中没有携带某个参数信息，则可能原因为：
 - A. 该参数值为默认值；
 - B. 该参数不存在；
 - C. 不支持查询该参数。

2.1.1.1 查询设备信息

该 API 用于查询 OM 设备自身的相关信息，如：生产商、硬件版本、软件版本，以及所有的分机、分机组和中继等。

请求示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Query">
```

```
<DeviceInfo/>
</Control>
```

参数说明:

参数名称	类型	参数说明	参数值说明
<DeviceInfo>	string	OM 设备的基本信息	无值

响应示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<DeviceInfo>
  <manufacturer>New Rock Technologies, Inc</manufacturer>
  <model>Rev 6.0.0 OM20-2S/2</model>
  <version>Rev 2.1.5.111</version>
  <mac>00:0E:A9:00:12:BD </mac>
  <devices>
    <ext lineid="Phone 1" id="200" />
    <ext lineid="IPPhone 50" id="208" />
    <line lineid="Line 2" id="02161208234" />
    <line lineid="IPLine 21" id="02161204000" />
    <group id="1">
      <ext id="200" />
    </group>
  </devices>
</DeviceInfo>
```

参数说明:

<>表示必选项, []表示可选项 (当参数值为默认值或空时, 响应消息可能不携带该参数), | 表示或者关系。

参数名称	类型	参数说明	参数值说明
DeviceInfo	object	OM 设备信息	无
<manufactu rer>	string	生产商	New Rock Technologies, Inc
<model>	string	硬件版本	如 Rev 6.0.0 OM20-2S/2

参数名称	类型	参数说明	参数值说明
<version>	string	软件版本	如 Rev 2.1.5.111
[mac]	string	物理地址	如 00:0E:A9:00:12:BD (由系统参数 <code>API_MAC</code> 决定是否携带 MAC 地址)
[ext]	object	分机	
<ext lineid>	string	分机的线路编号, 是分机的唯一固定标识	<Phone IPPhone> {NO.}, 如: Phone 1
<ext id>	string	分机号	
[line]	object	中继 (外线)	line 和 trunk 是指同一个对象, 即中继 (外线)
<line lineid>	string	中继的线路编号, 是中继的唯一固定标识	<Line IPLine> { NO.}, 如: Line 13
[line id]	string	中继号	
[group id]	int	分机组的序号	1~50

注:

- 1) 响应结果中包含所有的分机线路和中继线路信息, 如果设备线路量很大, 注意查询接收的缓存空间;
- 2) 设备线路量大时, 建议使用 web 的分页查询接口分别获取分机或中继线路。

2.1.1.2 查询分机

该 API 用于查询指定分机的相关信息, 如: 配置参数、分机状态、通话方等。

请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Query">
  <ext id="208"/>
</Control>
```

参数说明:

参数名称	类型	参数说明	参数值说明
<ext id>	string	分机号	必须为 OM 上的有效分机，值不能为空

响应示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Status>
  <ext id="208">
    <lineid>Phone 1</lineid>
    <group id="1"/>
    <group id="2"/>
    <staffid>1304081</staffid>
    <Call_Pickup>yes</Call_Pickup>
    <Fwd_Number>18603752801</Fwd_Number>
    <Call_Restriction>3</Call_Restriction>
    <Off_Line_Num>200</Off_Line_Num>
    <mobile>18603752800</mobile>
    <fork>18603752802</fork>
    <email>admin@hotmail.com</email>
    <record>on</record>
    <api>7</api>
    <voicefile>welcome</voicefile>
    <state>active</state>
    <outer id="8" from="208" to="13012345678" trunk="02161208234"
callid="28680">
      <state>talk</state>
    </outer>
  </ext>
</Status>
```

参数说明:

<>表示必选项, []表示可选项 (当参数值为默认值或空时, 响应消息可能不携带该参数), | 表示或者关系。

参数名称	类型	参数说明	参数值说明
<ext id>	string	分机号	纯数字字符串
<lineid>	string	分机的线路编号, 是分机的唯一固定标识	IPPhone Phone XXX
[staffid]	string	工号, 分机接通前会向来电方播放该工号	纯数字字符串
[group]	int	分机组, 这里为该分机所属的分机组	1~50
[voicefile]	string	语音文件, 这里为分机队列中排队等待时播放的语音文件	只支持.dat 和.pcm 格式
[email]	string	员工的电子邮件地址(暂无对应功能)	
[Call_Restriction]	int	呼叫权限	0: 内线, 1: 市话, 2: 国内, 3: 国际
[Call_Pickup]	string	代接权限, 是否允许来电被其它分机代接	Yes: 允许, no: 不允许
[No_Disturb]	string	免打扰功能开关, 开启免打扰后分机将屏蔽所有来电, 但能主动发起呼叫	on: 开启, off: 关闭
[Fwd_Type]	int	呼叫转移方式	0: 关闭, 1: 全转, 2: 遇忙或无应答转
[Fwd_Number]	string	呼叫转移号码	值为空时关闭
[fork]	string	同振号码	值为空时关闭
[mobile]	string	分机绑定的手机号, 该手机号可作为呼叫转移、离线转移的缺省配置	值为空时关闭
[record]	string	实时录音功能开关	on: 开启, off: 关闭

参数名称	类型	参数说明	参数值说明
[api]	int	API 的功能开关	0:关闭 API 状态监控 7: 开启 API 状态监控 点击查看 API 功能开关详细介绍
[state]	string	线路状态	Ready: 空闲可用 Active: 振铃、回铃或通话中 Progress: 模拟分机摘机后等待拨号以及拨号过程中 Offline: IP 分机离线 Offhook: 模拟分机听催挂音时的状态
[outer]	object	去电, 这里作为该查询分机的通话方	
[id]	int	去电的编号,可依据该参数进行呼叫转接	
[from]	string	原始主叫号码	
<to>	string	原始被叫号码 (对于 visitor 而言, 原始被叫为来电呼入中继号码)	
[trunk]	string	中继号, 即该去电从该中继呼出	
[callid]	int	通话的相对唯一标识符	
[state]	string	通话状态	Talk: 通话进行中 Progress: 呼叫处理过程中 Wait: 呼叫等待中
[visitor]	object	来电, 这里作为该查询分机的通话方	

参数名称	类型	参数说明	参数值说明
<id>	int	来电的编号, 可依据该参数进行呼叫转接等操作	
<from>	string	原始主叫号码	
<to>	string	原始被叫号码 (对于 visitor 而言, 原始被叫为来电呼入的中继号码)	
<callid>	int	通话的相对唯一标识符	
[state]	string	通话状态	Talk: 通话进行中 Progress: 呼叫处理过程中 Wait: 呼叫等待中
[ext]	object	分机, 这里作为该查询分机的通话方	
<id>	string	分机号	
[state]	string	通话状态	Talk: 通话进行中 Progress: 呼叫处理过程中 Wait: 呼叫等待中

2.1.1.3 查询中继

该 API 用于查询指定中继 (又称为外线) 的相关信息, 如: 配置参数、线路状态、呼叫状态等。

请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Query">
  <trunk id="2174"/>
</Control>
```

参数说明:

参数名称	类型	参数说明	参数值说明
<trunk id>	string	中继号码 (外线号码)	必须为 OM 上的有效中继, 值不能为空

响应示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Status>
  <trunk id="2174">
    <lineid>Line 75</lineid>
    <state>active</state>
    <visitor id="2" from="202" to="2174" callid="36866">
      <state>talk</state>
    </visitor>
  </trunk>
</Status>
```

参数说明:

<>表示必选项, []表示可选项 (当参数值为默认值或空时, 响应消息可能不携带该参数), | 表示或者关系。

参数名称	类型	参数说明	参数值说明
<trunk id>	string	中继 (外线) 号	数字字符串
<lineid>	string	中继的线路编号, 是中继的唯一固定标识	IPLine Line XXX
<state>	string	中继的线路状态	ready: 可用 active: 摘机、振铃或通话中 unwired: 未接线 offline: 离线 注: 对于 IP 中继多路并发时, 只要有一路空闲可用, 其状态就是 ready。
[outer]	object	去电	
<id>	int	去电的编号, 可通过该	

参数名称	类型	参数说明	参数值说明
		参数对去电进行转接、查询、挂断等操作	
<from>	string	原始主叫号码	
<to>	string	原始被叫号码（对于 visitor 而言，原始被叫为来电呼入中继号码）	
<trunk>	string	中继号，该去电通过该中继呼出	
<callid>	int	通话的相对唯一符	
<state>	string	通话状态	Talk: 通话进行中 Progress: 呼叫处理过程中 Wait: 呼叫等待中
[visitor]	object	来电	
<id>	int	来电的编号，可通过该参数对来电进行转接、查询、挂断等操作	
<from>	string	原始主叫号码	
<to>	string	原始被叫号码（对于 visitor 而言，原始被叫为来电呼入的中继号码）	
<callid>	int	该路通话的相对唯一的编号	
<state>	string	通话状态	Talk: 通话进行中 Progress: 呼叫处理过程中 Wait: 呼叫等待中

2.1.1.4 查询来电

该 API 用于查询指定来电的相关信息，如：来电的属性参数(编号、原始主叫、原始被叫、通话状态、相对唯一标识符)、来电的通话方、呼叫状态。

请求示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Query">
  <visitor id="1"/>
</Control>
```

参数说明：

参数名称	类型	参数说明	参数值说明
<visitor id>	int	来电的编号	数字，值为空时列举所有来电

响应示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Status>
  <visitor id="1" from="02167103750" to="02161208234" callid="49189">
    <ext id=" 200" />
    <state> progress </state>
  </visitor>
</Status>
```

注：来电 1 是由外部电话 02167103750 通过中继线 02161208234 呼入到 OM 设备，并且当前正在和分机 200 通话中。

参数说明：

<>表示必选项，[]表示可选项（当参数值为默认值或空时，响应消息可能不携带该参数），| 表示或者关系。

参数名称	类型	说明
[visitor]	object	来电，为由中继呼入 OM 的外部通话
<id>	int	来电的编号，可依据该参数对来电进行转接、查询、挂断等操作
<from>	string	原始主叫号码

参数名称	类型	说明
<to>	string	原始被叫号码（对于 visitor 而言，原始被叫为来电呼入的中继号码）
<callid>	int	通话的相对唯一标识符
[ext menu outer]	object	来电的通话方，可能为分机、语音菜单、呼叫寄存区、广播区、去电，为空时表明来电呼入 OM 后尚未被转接
[state]	string	Talk: 通话进行中 Progress: 呼叫处理过程中 Wait: 呼叫等待中

2.1.1.5 查询去电

该 API 用于查询指定去电的相关信息，如：去电的编号、主叫方、被叫方、通过的中继号码、通话状态以及通话的相对唯一标识符。

请求示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Query">
  <outer id="5"/>
</Control>
```

参数说明：

参数名称	类型	参数说明	参数值说明
<outer id>	int	去电的编号	数字，值为空时列举所有去电

响应示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Status>
  <outer id="5" from="200" to="13012345678" trunk="02161208234" callid="32773">
    <outer id="5" from="200" to="13012345678" trunk="02161208234"
callid="32773"/>
  </outer>
</Status>
```

解释： 去电 5 是由分机 200 对外部电话 13012345678 发起的呼叫，该路通话所经过的中继线为 02161208234。

参数说明:

<>表示必选项, []表示可选项 (当参数值为默认值或空时, 响应消息可能不携带该参数)。

参数名称	类型	说明
[outer]	object	去电, 从 OM 设备的中继呼出的通话
<id>	int	去电的编号,可依据该参数进行转接、挂断、查询等操作
<from>	string	原始主叫号码
<to>	string	原始被叫号码 (对于 visitor 而言, 原始被叫为来电呼入的中继号码)
<trunk>	string	中继号, 这里为该去电呼出时通过的中继
<callid>	int	通话的相对唯一标识符
[state]	string	通话状态, Talk: 通话进行中 Progress: 呼叫处理过程中 Wait: 呼叫等待中

2.1.1.6 查询分机组

该 API 用于查询分机组的相关信息, 如: 配置参数 (分机成员、呼叫排队时播放的背景音乐、呼叫分配规则)、正在该分机组队列中等待的来电。

注: 分机组队列中的呼叫排队顺序为从上到下。

请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Query">
  <group id="1"/>
</Control>
```

参数说明:

参数名称	类型	参数说明	参数值说明
<group id>	int	分机组的编号	1~50, 值为空时列举所有分机组

响应示例:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<Status>
  <group id="1">
    <ext id="200"/>
    <ext id="208"/>
    <voicefile>NowMorning</voicefile>
    <distribution>sequential</distribution>
    <visitor id="27" from="02167103750" to="02161208234" callid="49162"/>
    <visitor id="28" from="13012345678" to="02161204000" callid="49164"/>
  </group>
</Status>

```

解释：分机组 1 内有两个分机成员（分机 200 和分机 208），当来电转接到该分机组后，分机组按照 **distribution**（呼叫分配规则）的类型将呼叫分配给分机。当组内分机全忙，来电将会在分机组队列中等待，等待音乐为 **voicefile** 指定的语音文件。

参数说明：

<>表示必选项，[]表示可选项（当参数值为默认值或空时，响应消息可能不携带该参数）。

参数名称	类型	说明
<group id>	int	分机组的序号
[voicefile]	int	语音文件，支持 dat 和 pcm 两种格式，这里为呼叫等待时播放的音乐
[distribution]	string	有效值： sequential （顺选）、 circular （轮选）、 group （群振），默认值： circular
[ext]	string	分机，这里为该分机组中的分机成员
[visitor]	object	来电，这里为正处于该分机组等待队列中的来电
<id>	int	来电的编号，可依据该参数对来电进行转接、挂断、查询等操作
<from>	string	原始主叫号码
<to>	string	原始被叫号码（对于 visitor 而言，原始被叫为来电呼入的中继号码）
<callid>	string	通话的相对唯一标识符

2.1.1.7 查询语音菜单

该 API 用于查询语音菜单的相关信息，如：配置参数（语音文件、拨号检测长度、按键检查结束符）、转接到该菜单的呼叫信息等。

请求示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Query">
  <menu id="1"/>
</Control>
```

参数说明：

参数名称	类型	参数说明	参数值说明
<menu id>	int	语音菜单的编号	1~50, 值为空表示列举所有语音菜单

响应示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Status>
  <menu id="1">
    <voicefile>welcome</voicefile>
    <repeat>3</repeat>
    <infolength>5</infolength>
    <exit>#</exit>
    <outer id="44" from="200" to="02167103750" trunk="02161208234"/>
    <visitor id="46" from="13012345678" to="02161204000"/>
  </menu>
</Status>
```

注：来电 13012345678 和去电 02167103750 被转接到语音菜单 1 内，该菜单向它们播放名称为 **welcome** 的语音文件。另外，如果通话方的拨号按键长度达到 5，或者输入了 # 号时，则 OM 会将该按键信息封装在 [DTMF 事件报告](#) 中推送给应用服务器。如果当前没有任何呼叫被转接到该菜单，响应包体中将携带 XML 注释字段：<!-- Empty Waiting List -->，属正常现象。

参数说明：

<> 表示必选项，[] 表示可选项（当参数值为默认值或空时，响应消息可能不携带该参数）。

参数名称	类型	说明
------	----	----

参数名称	类型	说明
<menu id>	int	语音菜单的编号，用于查询、配置、转接等操作的判断依据。
[voicefile]	string	语音文件，当呼叫转接到该菜单并接通后，OM 会向通话方播放该文件。（只支持 dat 和 pcm 格式。）
[repeat]	int	语音文件的播放次数，取值范围 0~50，值为 0 时循环播放。
[infolength]	int	拨号检测长度，当按键长度达到该长度时，OM 则将已统计到的按键信息（DTMF 事件）推送给应用服务器，并重新开始统计。
[exit]	char	按键检查结束符，当该菜单的通话方一旦拨了该字符后，OM 会立刻将已统计到的按键信息（DTMF 事件）推送给应用服务器，并重新开始统计。
[visitor]	object	来电（从中继呼入的通话），这里指被转接到该语音菜单中的来电
<id>	int	来电的编号：用于查询、转接、挂断等操作的判断依据。
<from>	string	原始主叫号码
<to>	string	原始被叫号码（对于 visitor 而言，原始被叫为该 visitor 呼入 OM 的中继）
<callid>	int	通话的相对唯一标识符（注：其值会循环利用，不可作为永久标识）
[outer]	object	去电(从中继呼出的通话)，这里指被转接到该语音菜单中的去电
<id>	int	去电的编号，用于查询、转接、挂断等操作的判断依据。
<from>	string	原始主叫号码
<to>	string	原始被叫号码（对于 visitor 而言，原始被叫为该 visitor 呼入 OM 的中继）
<trunk>	string	中继号，这里指该路通话通过该中继线路呼出
<callid>	int	通话的相对唯一标识符（注：其值会循环利用，不可作为永久标识）

2.1.2 配置(Assign)

此类 API 用于对 OM 的对象进行参数配置。对象包括：语音菜单（menu）、分机（ext）、分机组(group)。

配置请求的规则说明：

- 1) **分机的配置说明：**分机的各个配置参数之间是相互独立，即，配置命令中如果没有携带某参数，则该参数保持原值不变；
- 2) **其他对象的配置说明：**除了分机，其他对象的配置参数都是非独立的，即，如果配置命令中没有携带某参数，则该参数值有可能被清空或还原默认值；
- 3) **特别注意：**API 配置不会写配置文件，而是缓存在内存，设备重启会丢失，所以应用服务器应该保留做过的相关配置，当收到设备的**重启事件（BOOTUP）**时，将保留的配置重新下发一次。

配置结果的规则说明：

- 1) 如果参数值为默认值或空时，响应结果中可能不携带该参数；
- 2) 如果配置参数不生效时，响应结果中也可能不携带该参数；
- 3) 除此之外，配置结果中将尽可能包含所有可提供的相关参数和状态信息等。

2.1.2.1 配置分机

该 API 用于对指定线路号的分机进行参数配置，并返回配置后的结果及状态。

配置参数包括：分机号、呼叫状态方式、呼叫权限、免打扰、代接权限等。

请求示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
  <ext lineid="IPPhone 50">
    <id>208</id>
    <staffid>1304081</staffid>
    <mobile>18603752800</mobile>
    <group>1</group>
    <group>2</group>
    <voicefile>busy</voicefile>
    <email>admin@hotmail.com</email>
    <Call_Restriction>3</Call_Restriction>
    <Call_Pickup>no</Call_Pickup>
```

```

<No_Disturb>no</No_Disturb>
<Fwd_Type>0</Fwd_Type>
<Fwd_Number>18603752801</Fwd_Number>
<Fork>18603752802</Fork>
<WAKEUP>06:30</WAKEUP>
<record>on</record>
<autoAnswer>yes </autoAnswer>
<api>7</api>
</ext>
</Control>

```

参数说明:

<>表示必选项, []表示可选项, (当参数值为默认值或空时, 响应消息可能不携带该参数), | 表示或者关系。

参数名称	类型	参数说明	参数值说明
<ext lineid>	string	分机的线路编号 注: 通过查询设备信息可获取线路编号	<Phone IPPhone> {number}, 如 Phone 1
[id]	string	分机号码	纯数字非空字符串
[staffid]	string	工号, 分机接通前会向来电方播放该工号	纯数字字符串, 值为空时不生效
[group]	int	分机所属的分机组, 同一分机可属多个分机组。 注: 若配置分机时未携带 group 信息, 则分机将从该分机组中被清除。	1~50
[voicefile]	string	语音文件, 这里为分机队列中排队等待时播放的语音文件	只支持.dat 和.pcm 格式 点击查看语音文件的配置说明
[email]	string	员工的电子邮件地址 (暂无对应功能)	服从标准邮箱格式
[Call_Restriction]	int	呼叫权限	0: 内线, 1: 市话 2: 国内, 3: 国际

参数名称	类型	参数说明	参数值说明
[Call_Pickup]	string	代接权限，是否允许来电被其它分机代接	Yes:允许 no:不允许
[No_Disturb]	string	免打扰功能开关（ 注： 该分机为总机时无效），开启免打扰后分机将屏蔽所有来电，但能主动发起呼叫	on: 开启 off: 关闭
[Fwd_Type]	int	呼叫转移方式(注： 该分机为总机时无效)	0: 关闭，1: 全转 2: 遇忙或无应答转
[Fwd_Number]	string	呼叫转移号码（ 注： 该分机为总机时无效）	值为空时关闭
[Fork]	string	同振号码	值为空时关闭
[mobile]	string	分机绑定的手机号，该手机号可作为分机同振、或呼叫转移时的缺省配置	值为空时关闭
[WAKEUP]	string	叫醒闹钟时间，24 小时制，精确到分钟。（该功能相对局限，可通过其他方式扩展）	格式为 HH:MM，值为 00:00 时关闭闹铃
[record]	string	实时录音功能开关	on: 开启，off: 关闭，默认值为 off 点击查看录音配置和管理
[api]	int	API 的功能开关	0:关闭 API 状态监控 7: 开启 API 状态监控 点击查看更多
[autoAnswer]	string	IP 分机点击拨号是否自动应答(注： 该参数仅对 IP 分机有效，且需要 IP 话机支持。如，迅时、方位等大	yes:是，no:否 默认值为 no

参数名称	类型	参数说明	参数值说明
		部分品牌的话机)	

响应示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Status>
  <ext id="208">
    <id>208</id>
    <staffid>1304081</staffid>
    <mobile>18603752800</mobile>
    <group>1</group>
    <group>2</group>
    <voicefile>busy</voicefile>
    <email>admin@hotmail.com</email>
    <Call_Restriction>3</Call_Restriction>
    <Call_Pickup>no</Call_Pickup>
    <No_Disturb>off</ No_Disturb>
    <Fwd_Type>0</Fwd_Type>
    <Fwd_Number>18603752801</Fwd_Number>
    <Fork>18603752802</Fork>
    <Off_Line_Num>218</Off_Line_Num>
    <WAKEUP>06:30</WAKEUP>
    <record>on</record>
    <autoAnswer>yes </autoAnswer>
    <api>7</api>
  </ext>
</Status>
```

参数说明:

略, 可参考查询分机的 API 响应消息中的参数说明。

(友情提示: 超链接返回的快捷键为 Alt + ←)

2.1.2.2 配置分机组

该 API 用于对指定编号的分机组进行参数配置, 并返回配置后的结果及状态。

配置参数包括: 分机成员、语音文件、呼叫分配规则等。

注: 一个分机组最多支持 20 个分机成员。

请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
  <group id="1">
    <voicefile>NewMorning</voicefile>
    <distribution>sequential</distribution>
    <ext>200</ext>
    <ext>208</ext>
  </group>
</Control>
```

参数说明:

<>表示必选项, []表示可选项, (当参数值为默认值或空时, 响应消息可能不携带该参数)。

参数名称	类型	参数说明	参数值说明
<group id>	int	分机组的编号	1~50
[distribution]	string	呼叫分配规则	有效值: sequential(顺选) group (群振) circular (轮选) 默认值: circular
[ext]	string	分机号码, 这里为分机组成员	必须为有效分机

响应示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Status>
  <group id="1">
    <voicefile>NewMorning</voicefile>
    <ext id="200"/>
    <ext id="208"/>
    <!-- Empty Waiting List -->
  </group>
</Status>
```

解释: 其中<!-- Empty Waiting List -->表明该分机组队列中没有来电在排队等待。

特别注意:

- 1) OM 系统重启后，分机组内的分机被清除，须重新配置分机组，可根据系统启动事件 BOOTUP 来判断。
- 2) 配置分机组时，只会累加分机不会删除组内之前的分机，建议配置分机组前先清空该分机组。

清空分机组的命令：

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
  <group id="1">
  </group>
</Control>
```

参数说明：

略，可参考查询分机组的 API 响应消息中的参数说明。

（友情提示：超链接返回的快捷键为 Alt + ←）

2.1.2.3 配置语音菜单

该 API 用于对指定编号的语音菜单进行参数配置，并返回配置后的结果及状态。

请求示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
  <menu id="1">
    <voicefile>welcome</voicefile>
    <repeat>3</repeat>
    <infolength>5</infolength>
    <exit>#</exit>
  </menu>
</Control>
```

参数说明：

<>表示必选项，[]表示可选项，（当参数值为默认值或空时，响应消息可能不携带该参数）。

参数名称	类型	参数说明	参数值说明
------	----	------	-------

参数名称	类型	参数说明	参数值说明
<menu id>	int	语音菜单的编号	有效值 1~50
[voicefile]	string	语音文件，当有呼叫被转接到该菜单时播放该语音文件。	只支持 dat 和 pcm 格式，可指定远程下载路径。 点击查看语音文件配置说明
[repeat]	int	语音文件的播放次数	有效值 0~65535，0 为循环播放，默认值为 0
[infolength]	int	按键检查长度，当被转接到该菜单内的通话方输入按键的长度达到该长度时，OM 会将已检测到的按键信息推送给应用服务器。	合法值为 1~255（一字节长度）默认值为 1
[exit]	char	按键检查结束符，当被转接到该菜单内的通话方按键输入该符号后，OM 会立刻将已检测到的按键信息推送给应用服务器。	合法值：1~9、A~D、*、#，默认为空

响应示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Status>
  <menu id="1">
    <voicefile> welcome </voicefile>
    <repeat>3</repeat>
    <infolength>5</infolength>
    <exit>#</exit>
    <outer id="44" from="200" to="02167103750" trunk="02161208234"/>
    <visitor id="46" from="13012345678" to="02161204000"/>
  </menu>
</Status>
```

解释： 其中 outer 和 visitor 为当前呼叫驻留在该语音菜单中的去电和来电。

注：若当前没有任何呼叫被转接到该菜单，响应包体中将携带 XML 注释字段：<!-- Empty Waiting List -->。

参数说明：

略，可参考查询语音菜单的 API 响应消息中的参数说明。

（友情提示：超链接返回的快捷键为 Alt + ←）

2.1.3 呼叫保持和接回(Hold/Unhold)

通话过程中，可通过 Hold 命令使对方保持，通过 Unhold 命令解除保持。

2.1.3.1 呼叫保持 (Hold)

该 API 用于将分机的当前通话呼叫保持。保持后，分机可以通过 API 发起新的呼叫。

如需恢复通话，可使用呼叫接回命令 (Unhold)。另外，分机和新的通话结束后会自动接回原通话。

请求示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Hold">
    <ext id="200"/>
</Control>
```

参数说明：

参数名称	类型	说明
<ext id>	string	分机号码

响应示例：

```
HTTP/1.0 200 OK
Date: Wed, 22 Feb 2017 08:04:44 GMT
Server:
X-Frame-Options:DENY
Cache-Control:no-cache
Pragma:no-cache
Expires:-1
Access-Control-Allow-Headers: accept, Content-Type
Access-Control-Allow-Origin: (null)
```

Connection: close

Content-Length: 0

<!--包体为空-->

2.1.3.2 呼叫接回 (Unhold)

使用该 API 能够将分机呼叫被保持 (Hold) 的通话重新接回。

请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Unhold">
  <ext id="200"/>
</Control>
```

参数说明:

参数名称	类型	说明
<ext id>	string	分机号码

响应示例:

HTTP/1.0 200 OK

Date: Wed, 22 Feb 2017 08:04:44 GMT

Server:

X-Frame-Options:DENY

Cache-Control:no-cache

Pragma:no-cache

Expires:-1

Access-Control-Allow-Headers: accept, Content-Type

Access-Control-Allow-Origin: (null)

Connection: close

Content-Length: 0

<!--包体为空-->

2.1.4 静音开启与解除(Mute/Unmute)

若通话过程中, 不想让对方听到你的声音, 可通过静音命令使对方静音, 通过解除静音命令, 使通话恢复正常。

2.1.4.1 开启静音(Mute)

在分机通话时，可利用该 API 将其话筒静音。即，分机的通话方不能听到分机的声音，而分机可以听到对方的声音。

如需解除静音，可使用解除静音命令（Unmute），否则通话结束后也会自动解除静音。

请求示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Mute">
  <ext id="200"/>
</Control>
```

参数说明：

参数名称	类型	说明
<ext id>	string	分机号码

响应示例：

```
HTTP/1.0 200 OK
Date: Wed, 22 Feb 2017 08:04:44 GMT
Server:
X-Frame-Options:DENY
Cache-Control:no-cache
Pragma:no-cache
Expires:-1
Access-Control-Allow-Headers: accept, Content-Type
Access-Control-Allow-Origin: (null)
Connection: close
Content-Length: 0
<!-- 包体为空-->
```

2.1.4.2 解除静音(Unmute)

该 API 用于将执行静音后的分机恢复，使该分机的通话方重新听到声音。

执行静音后如果未执行解除静音，则本次通话结束后也会自动解除，不影响下一次通话。

请求示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Unmute">
```

```
<ext id="200"/>
</Control>
```

参数说明:

参数名称	类型	说明
<ext id>	string	分机号码

响应示例:

```
HTTP/1.0 200 OK
Date: Wed, 22 Feb 2017 08:04:44 GMT
Server:
X-Frame-Options:DENY
Cache-Control:no-cache
Pragma:no-cache
Expires:-1
Access-Control-Allow-Headers: accept, Content-Type
Access-Control-Allow-Origin: (null)
Connection: close
Content-Length: 0
<!--包体为空-->
```

2.1.5 监听和插播(Monitor/Talk/Listen)

2.1.5.1 监听(Monitor)

该 API 用于实现一个分机监听另一个分机的当前通话。

操作步骤:

- 1) 已知: 分机 A 当前空闲, 分机 B 正在通话中;
- 2) 执行分机 A 监听分机 B 的 API;
- 3) 执行成功时, 分机 A 振铃, 摘机后即可监听到分机 B 及其通话方之间的通话内容。

监听的注意事项:

- 1) 监听方必须开启“监听”开关;
- 2) 被监听方和被监听方的通话方 (另一个分机) 必须关闭“禁止被监听”开关;
- 3) 一个分机最多同时只能被一个分机监听;
- 4) 如果分机在进行三方通话时, 则不能被监听;

- 5) 监听范围：只限于同一台 OM 设备上的分机相互监听，暂不支持组网的 OM 设备之间相互监听；
- 6) 监听方挂机，不影响原有通话。

监听和禁止被监听开关

登录 OM 的 web 页面>分机设置>模拟分机/IP 分机>设置：



请求示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Monitor">
  <ext id="200"/>
  <ext id="208"/>
</Control>
```

参数说明：

参数名称	类型	说明
<ext id>	string	分机号码，第一个为监听方，第二个为被监听方

响应示例：

HTTP/1.0 200 OK
Date: Wed, 22 Feb 2017 08:04:44 GMT

```
Server:
X-Frame-Options:DENY
Cache-Control:no-cache
Pragma:no-cache
Expires:-1
Access-Control-Allow-Headers: accept, Content-Type
Access-Control-Allow-Origin: (null)
Connection: close
Content-Length: 0
<!--包体为空-->
```

2.1.5.2 从监听到插播状态变换 (Talk)

该 API 用于使分机从监听状态切换到插播状态。

限制条件:

版本要求: Rev2.1.5.119 及以上。

操作步骤:

- 1) 已知: 分机 A 在监听分机 B 与其通话方的通话;
- 2) 执行分机 A 的从监听到插播状态变换的 API;
- 3) 执行成功时, 分机 A 与分机 B 建立通话, 分机 B 的原通话方听保持音。

请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Talk">
    <ext id="201"/>
</Control>
```

参数说明:

参数名称	类型	说明
<ext id>	string	插播方分机号码

响应示例:

```
HTTP/1.0 200 OK
Date: Wed, 22 Feb 2017 08:04:44 GMT
Server:
X-Frame-Options:DENY
```

```

Cache-Control:no-cache
Pragma:no-cache
Expires:-1
Access-Control-Allow-Headers: accept, Content-Type
Access-Control-Allow-Origin: (null)
Connection: close
Content-Length: 0
<!-- 包体为空-->

```

2.1.5.3 从插播到监听状态变换 (Listen)

该 API 用于使分机从插播状态切换到监听状态。

限制条件:

版本要求: Rev2.1.5.119 及以上。

操作步骤:

- 1) 已知: 分机 A 在插播分机 B 的通话;
- 2) 执行分机 A 的从插播到监听状态变换的 API;
- 3) 执行成功时, 分机 A 监听分机 B 及其原通话方的通话。

请求示例:

```

<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Listen">
  <ext id="201"/>
</Control>

```

参数说明:

参数名称	类型	说明
<ext id>	string	监听方分机号码

响应示例:

```

HTTP/1.0 200 OK
Date: Wed, 22 Feb 2017 08:04:44 GMT
Server:
X-Frame-Options:DENY
Cache-Control:no-cache
Pragma:no-cache

```

Expires:-1

Access-Control-Allow-Headers: accept, Content-Type

Access-Control-Allow-Origin: (null)

Connection: close

Content-Length: 0

<!-- 包体为空-->

2.1.6 强插(Bargein)

该 API 用于让一个分机强制插入到另一个分机的通话中去，从而形成三方通话。

操作步骤:

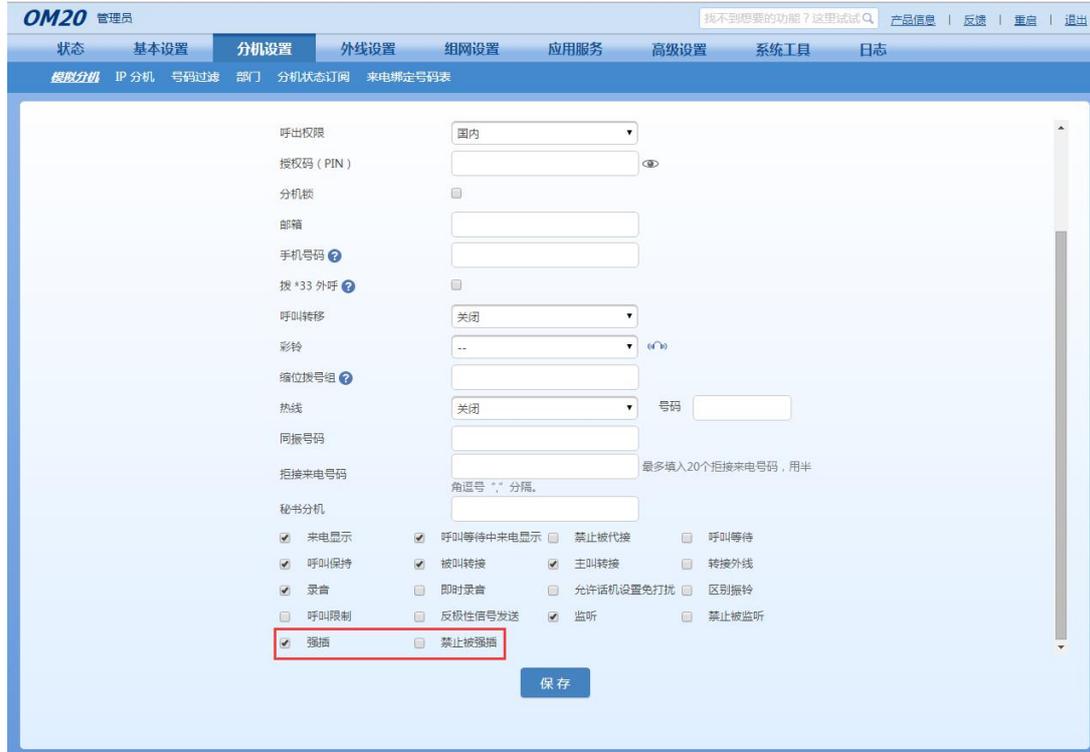
- 1) 已知：分机 A 当前空闲，分机 B 正在通话中；
- 2) 执行分机 A 强插分机 B 的 API；
- 3) 执行成功时，分机 A 振铃，摘机后即可形成三方通话。

强插的注意事项:

- 1) 强插方必须开启“强插”开关；
- 2) 被强插方和被强插方的通话方（另一个分机）必须关闭“禁止被强插”开关；
- 3) 如果分机在进行三方会议时，则不能被强插；
- 4) 强插范围：只限于同一台 OM 设备上的分机相互强插，暂不支持组网的 OM 设备之间相互强插；任何一分机挂机，都不影响其他两分机通话；
- 5) 强插方退出三方通话后，可再次执行强插进入三方通话；而对于被强插方及其通话方，退出后不可再次强插进入通话。

强插和禁止被强插开关

登录 OM 的 web 页面 > 分机设置 > 模拟分机/IP 分机 > 设置:



请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Bargein">
  <ext id="200"/>
  <ext id="208"/>
</Control>
```

参数说明:

参数名称	类型	说明
<ext id>	string	分机号码, 第一个为强插方, 第二个为被强插方

响应示例:

```
HTTP/1.0 200 OK
Date: Wed, 22 Feb 2017 08:04:44 GMT
Server:
X-Frame-Options:DENY
Cache-Control:no-cache
```

Pragma:no-cache

Expires:-1

Access-Control-Allow-Headers: accept, Content-Type

Access-Control-Allow-Origin: (null)

Connection: close

Content-Length: 0

<!-- 包体为空 -->

2.1.7 强拆(Clear)

此类 API 用于强制挂断 OM 上的指定通话。该 API 可强拆的对象包括：分机、来电、去电。

强拆请求的规则说明：

- 1) 对于分机，强拆的判断依据是：ext id;
- 2) 对于来去电，强拆的判断依据是：visitor/outer id;
- 3) 双方通话时，不管强拆其中哪一方，该路通话都会被释放。

请求示例：

强拆分机：

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Clear">
  <ext id="200"/>
</Control>
```

强拆来电：

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Clear">
  <visitor id="2"/>
</Control>
```

强拆去电：

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Clear">
  <outer id="20"/>
</Control>
```

参数说明：

参数名称	类型	说明
<ext id>	String	分机号码（可通过 查询设备信息 获取所有分机号及其对应的线路号）
<visitor id>	int	来电的编号
<outer id>	int	去电的编号

响应示例：

```
HTTP/1.0 200 OK
Date: Wed, 22 Feb 2017 08:04:44 GMT
Server:
X-Frame-Options:DENY
Cache-Control:no-cache
Pragma:no-cache
Expires:-1
Access-Control-Allow-Headers: accept, Content-Type
Access-Control-Allow-Origin: (null)
Connection: close
Content-Length: 0
<!-- 包体为空 -->
```

2.2 呼叫转接命令 (Transfer)

应用服务器可向 OM 发送 Transfer 命令，实现呼叫、转接、队列和三方通话。

2.2.1 连接 (Connect)

Connect 命令可实现呼叫（如分机呼分机、menu 外呼、双向外呼）和来去电转接等功能。

2.2.1.1 分机呼分机

该 API 用于将一个分机呼叫另一个分机，从而使两者建立通话。

请求示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <ext id="200"/>
  <ext id="208"/>
</Transfer>
```

参数说明:

参数名称	类型	说明
<ext id>	string	分机号码, 第一个为主叫分机, 第二个为被叫分机, 先呼主叫还是被叫有参数 API_CALLING 决定

响应示例:

```
HTTP/1.0 200 OK
Date: Wed, 22 Feb 2017 08:04:44 GMT
Server:
X-Frame-Options:DENY
Cache-Control:no-cache
Pragma:no-cache
Expires:-1
Access-Control-Allow-Headers: accept, Content-Type
Access-Control-Allow-Origin: (null)
Connection: close
Content-Length: 0
<!--包体为空-->
```

2.2.1.2 分机呼外部电话

该 API 用于让分机向外部电话发起呼叫, 从而使两者建立通话。

请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <ext id="208"/>
  <outer to="13012345678"/>
</Transfer>
```

指定中继外呼:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <ext id="208"/>
  <trunk id="50017"/>
  <outer to="13012345678"/>
</Transfer>
```

加前缀外呼:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <ext id="208"/>
  <outer to="9,13012345678"/>
</Transfer>
```

二次拨号:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <ext id="208"/>
  <outer to="13012345678,,1"/>
</Transfer>
```

参数说明:

参数名称	类型	说明
<ext id>	string	分机号码
<outer to>	string	呼出的被叫号码
[trunk id]	string	指定中继号码, 可以为 IP 中继或者模拟中继号码

注:

- 1) 加前缀外呼时, 第四行中 9 为前缀, 即指定从模拟外线或 IP 外线呼出, 此时需配置设备的拨号规则为“外线加前缀, 内线直拨”。
- 2) 二次拨号时, “,,1”表示呼叫接通后, 主叫自动按键“1”, “,,”后可再加逗号来延迟按键时间, 一个逗号延迟 1 秒 (目前版本 V114, 只有通过 IP 外线外呼支持, 通过模拟外线外呼不支持, 待后续修改)。

响应示例:

通过 IP 中继呼出时的响应:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="TRANSIENT">
  <outer id="66" from="208" to="13012345678" trunk="50017" callid="8258"/>
  <ext id="208"/>
</Event>
```

通过模拟中继呼出时的响应:

注：通过模拟中继呼出时，响应消息中没有 `outer id`、`trunk`、`callid`，但随后 OM 会再次向应用服务器推送一条“**TRANSIENT**”事件消息，该消息会将这些参数信息补充完整。

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="TRANSIENT">
  <outer from="208" to="13012345678"/>
  <ext id="208"/>
</Event>
```

参数说明：

略，可参考查询去电的 API 响应消息中的参数说明。

（友情提示：超链接返回的快捷键为 Alt + ←）

2.2.1.3 来电转分机

该 API 用于将来电转接给分机，从而使来电和分机建立通话。

提示音说明：

- 1) OM 将来电转接给分机之前，会先向来电方播放转接提示音“XXX，转接中请稍后”，如果该分机配置了员工号则 XXX 为员工号，否则为分机号；
- 2) 如果用户在执行该 API 时，觉得系统自带的转接提示音不能满足要求，可临时指定新的 voicefile，该 voicefile 仅在本次转接中生效。

请求示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <visitor id="14"/>
  <ext id="200"/>
</Transfer>
```

注：转接时，语音提示如“200 转接中，请稍后”。

指定临时播放语音文件：

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <visitor id="14"/>
  <ext id="200"/>
  <voicefile>silence+silence+connect</voicefile>
</Transfer>
```

注：转接时，播放的语音为“转接中请稍后”，不再播报分机号。

参数说明：

参数名称	类型	说明
<visitor id>	int	来电的编号，范围：1~4096，从通话开始到结束的时间内是唯一的，但会被循环利用，可作为临时唯一标识。可根据 INVITE 或 INCOMING 事件获取。
<ext id>	string	分机号
[voicefile]	string	语音文件，用于临时替换转接提示音。支持数字字符串和 dat、pcm 格式的语音文件，多个文件之间用加号“+”串联。
[AutoAnswer]	string	IP 分机自动应答参数。来电转分机时，在命令中携带此标签，可使 IP 分机自动应答。 注：只有 IP 分机才支持自动应答，模拟分机不支持。

注：<voicefile>指定语音文件个数小于等于 10，只要语音文件存在（可下载）就可以正常播放，超过 10 个后不再播放（V2.1.5.113.2 修复：从第十一个语音文件去匹配内部语音文件，如果匹配到，就会播放，匹配不到，就报错）。

响应示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="TRANSIENT">
  <visitor id="14" from="13012345678" to="02161208234" callid="49226" />
  <ext id="200" />
</Event>
```

参数说明：

略，可参考查询来电的 API 响应消息中的参数说明。

（友情提示：超链接返回的快捷键为 Alt + ←）

2.2.1.4 来电转外部电话

该 API 用于让来电向外部电话发起呼叫，从而实现两个外部电话能够以 OM 为中转站建立通话。

执行权限：

执行外呼之前，需开启外线的外转外权限。【[点击查看外转外权限开启方法](#)】

提示音说明:

- 1) 转外线在默认情况下没有转接提示音;
- 2) 如果用户需要转外线也和转分机一样有转接提示音,则可在该 API 中指定语音文件(只对本次呼叫有效)。

请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <visitor id="3" />
  <outer to="13012345678"/>
</Transfer>
```

透传主叫号码:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <visitor id="3" />
  <outer to="13012345678" display="18201008888" />
</Transfer>
```

注: 指定中继外呼和加前缀外呼命令,可参考[分机呼外部电话](#);临时播放语音文件命令,可参考[来电转分机](#)。

参数说明:

参数名称	类型	说明
<visitor id>	int	来电的编号
<outer to>	string	要呼叫的外部电话号码
[display]	string	来电号码,用来透传主叫号码,使去电方的来电显示号码为实际来电号码。

注:

- 1) 采用 display 字段透传主叫号码时,须通过 web URL 配置参数 [API_DISPLAY\(id624\)](#) 的值为 yes;
- 2) 当有来电打入中继时,可根据 INCOMING 事件获取来电号码,再将获取到的号码实时添加到 display 字段中。

响应示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="TRANSIENT">
  <visitor id="46" from="13012345678" to="02161204000" callid="41006"/>
  <outer to="02167103750"/>
</Event>
```

参数说明:

略, 可参考查询来电的 API 响应消息中的参数说明。

(友情提示: 超链接返回的快捷键为 Alt + ←)

2.2.1.5 来电转语音菜单

该 API 用于将来电转接到语音菜单 (menu), 从而实现:

- 1) 向通话方播放语音文件。其中, 语音文件的播放次数和内容都可以控制。
- 2) 能够检查通话方输入的按键信息, 且按键满足 menu 配置的按键汇报条件后, OM 会将按键信息汇报给应用服务器。

播放临时语音文件:

- 1) 如呼叫转接 menu 时临时指定新的 voicefile, 则 OM 不再向通话方播放 menu 配置的 voicefile, 而是播放临时指定的 voicefile, 且仅播放一遍, 播放完毕汇报 EndOfAnn 事件;
- 2) 但是, DTMF 的汇报规则依然遵循 menu 的配置。

播放临时语音文件的应用场景举例:

- 1) **按键纠错提示:** 当用户在某个 menu 中按键不合规范时, 可重新将呼叫接回到该 menu, 并同时指定播放纠错提示音, 而非播放 menu 配置的 voicefile;
- 2) **播报动态语音:** 如, 余额、工单等内容不固定的菜单。

请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <visitor id="41" />
  <menu id="1"/>
</Transfer>
```

注: 临时播放语音文件命令, 可参考来电转分机。

参数说明:

参数名称	类型	说明
<visitor id>	int	来电的编号
<menu id>	int	语音菜单的编号

响应示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="TRANSIENT">
  <visitor id="41" from="02167103750" to="02161204000" callid="41001"/>
  <menu id="1" />
</Event>
```

参数说明:

略, 可参考查询来电的 API 响应消息中的参数说明。

(友情提示: 超链接返回的快捷键为 Alt + ←)

2.2.1.6 去电转分机

该 API 用于将去电 (已从 OM 呼出的通话) 转接给分机, 从而使两者能够建立通话。

请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <outer id="14"/>
  <ext id="200"/>
</Transfer>
```

参数说明:

参数名称	类型	说明
<outer id>	int	去电的编号
<ext id>	string	分机号码

响应示例:

```
<Event attribute="TRANSIENT">
```

```
<outer id="14" from="200" to="13012345678" trunk="61208111" callid="8258"/>
</Event>
```

参数说明:

略，可参考查询去电的 API 响应消息中的参数说明。

(友情提示: 超链接返回的快捷键为 Alt + ←)

2.2.1.7 去电转外部电话

该 API 用于让去电 (已从 OM 呼出的通话) 向外部电话发起呼叫, 从而实现两个外部电话能够以 OM 为中转站建立通话。

执行权限:

执行外呼之前, 需要开启外线的外转外权限。【[点击查看外转外权限开启方法](#)】

提示音说明:

- 1) 转外线时, 默认情况下没有转接提示音;
- 2) 如果用户需要转外线也和转分机一样有转接提示音, 则可在该 API 中指定语音文件。

请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <outer id="14" />
  <outer to="13012345678"/>
</Transfer>
```

注: 指定中继外呼和加前缀外呼命令, 可参考分机呼外部电话; 临时播放语音文件命令, 可参考来电转分机。

参数说明:

参数名称	类型	说明
<outer id>	int	去电的编号
<outer to>	string	要呼叫的外部电话号码

响应示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="TRANSIENT">
```

```

    <outer id="14" from="200" to="02167103750" trunk="02161204000" callid="49174"
/>
    <outer to="13012345678"/>
</Event>

```

参数说明:

略，可参考查询去电的 API 响应消息中的参数说明。

(友情提示: 超链接返回的快捷键为 Alt + ←)

2.2.1.8 去电转语音菜单

该 API 用于将去电转接到语音菜单 (menu)，从而实现:

- 1) 向通话方播放语音文件。其中，语音文件的播放次数和内容都可以控制。
- 2) 能够检查通话方输入的按键信息，且按键满足 menu 配置的按键汇报条件后，OM 会将按键信息汇报给应用服务器。

播放临时语音文件:

- 1) 如呼叫转接 menu 时临时指定新的 voicefile，则 OM 不再向通话方播放 menu 配置的 voicefile，而播放该临时指定的 voicefile，且仅播放一遍，播放完毕汇报 EndOfAnn 事件;
- 2) 但是，DTMF 的汇报规则依然遵循 menu 的配置。

播放临时语音文件的应用场景举例:

按键纠错提示: 当用户在某个 menu 中按键不合规范时，可重新将呼叫接回到该 menu，并同时指定播放纠错提示音，而非播放 menu 配置的 voicefile;

播报动态语音: 如，余额、工单等内容不固定的菜单。

请求示例:

```

<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
    <outer id="41" />
    <menu id="1"/>
</Transfer>

```

注: 临时播放语音文件命令，可参考来电转分机。

参数说明:

参数名称	类型	说明
<outer id>	int	去电的编号
<menu id>	int	语音菜单的编号

响应示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="TRANSIENT">
  <outer id="14" from="200" to="02167103750" trunk="02161204000"
  callid="49174" />
  <menu id="1"/>
</Event >
```

参数说明:

略, 可参考查询去电的 API 响应消息中的参数说明。

(友情提示: 超链接返回的快捷键为 Alt + ←)

2.2.1.9 语音菜单呼分机

该 API 用于语音菜单 (menu) 呼分机, 从而实现:

- 1) 向分机播放语音文件。其中, 语音文件的播放次数和内容都可以控制。
- 2) 能够检查分机输入的按键信息, 且按键满足 menu 配置的按键汇报条件后, OM 会将按键信息汇报给应用服务器。

常见使用场景:

- 1) 酒店分机的叫醒服务;
- 2) 会议通知等。

请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <menu id="1"/>
  <ext id="200"/>
</Transfer>
```

参数说明:

参数名称	类型	说明
<menu id>	int	语音菜单的编号
<ext id>	string	分机的电话号码

响应示例:

```
HTTP/1.0 200 OK
Date: Wed, 22 Feb 2017 08:04:44 GMT
Server:
X-Frame-Options:DENY
Cache-Control:no-cache
Pragma:no-cache
Expires:-1
Access-Control-Allow-Headers: accept, Content-Type
Access-Control-Allow-Origin: (null)
Connection: close
Content-Length: 0
<!-- 包体为空-->
```

2.2.1.10 语音菜单呼外部电话

该 API 用于语音菜单（menu）向外部电话发起呼叫，从而实现：

- 1) 当外部电话接通后，向其播放语音文件。其中，语音文件的播放次数和内容都可以控制。
- 2) 能够检查外部电话输入的按键信息，且按键满足 menu 配置的按键汇报条件后，OM 会将按键信息汇报给应用服务器。

执行权限:

执行外呼之前，需要开启外线的外转外权限。【[点击查看外转外权限开启方法](#)】

常见使用场景:

- 1) 客服质量回拨评分；
- 2) 语音导航用户调查。

请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
```

```

<menu id="1"/>
<outer to="13012345678"/>
</Transfer>

```

注：指定中继外呼和加前缀外呼，参考[分机外呼](#)。

参数说明：

参数名称	类型	说明
<menu id>	int	语音菜单的编号
<outer to>	string	呼出的被叫号码

响应示例：

```

<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="TRANSIENT">
  <outer id="1" from="" to="13012345678" trunk="02161208234" callid="20481"/>
  <menu id="1"/>
</Event>

```

参数说明：

略，可参考查询去电的 API 响应消息中的参数说明。

（友情提示：超链接返回的快捷键为 Alt + ←）

2.2.1.11 双向外呼（回拨）

该 API 用于让 OM 依次向外发起两路呼叫，并以 OM 作为中间点建立连接，从而实现两个外部电话之间建立通话。

呼叫顺序：

先呼主叫，主叫摘机后听到回铃音，然后再呼被叫，被叫摘机后双方建立通话。

中继资源要求：

- 1) 至少要有两条空闲可用的中继资源；
- 2) 第一路外呼默认使用 IP 中继，可通过参数 `CB_TYPE` 配置调整为模拟中继；
- 3) 第二路外呼按照 OM 的外呼规则执行。

执行权限：

执行双向外呼之前，需要开启线路的外转外权限。【[点击查看外转外权限开启方法](#)】

请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <outer to="13012345678"/>
  <outer to="02167103750"/>
</Transfer>
```

指定中继外呼:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <outer to="13012345678" display="50018"/>
  <outer to="02167103750" display="50019"/>
</Transfer>
```

参数说明:

参数名称	类型	说明
<outer to>	string	呼出的外部电话号码, 第一个为主叫方, 第二个为被叫方
<display>	string	指定中继号码, 可以为 IP 中继或者模拟中继号码, 需要配置 API_DISPLAY 和 API_CPN 参数

响应示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="TRANSIENT">
  <outer id="3" from="" to="13012345678" trunk="02161208234" callid="45059"/>
</Event>
```

参数说明:

略, 可参考查询去电的 API 响应消息中的参数说明。

(友情提示: 超链接返回的快捷键为 Alt + ←)

2.2.1.12 语音插播

该 API 用于向正在通话的一方插播语音。执行插播命令后, 被插播方听语音, 其通话方静音, 语音播放完毕后自动恢复原有通话。

如果需要双向插播, 可同时执行两次命令。

限制条件:

版本: Rev 2.1.5.117 及以上

适合场景:

分机呼分机、来电转分机、分机外呼三种场景。

请求示例:

向分机插播语音:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <ext id="200" />
  <voicefile>welcome</voicefile>
</Transfer>
```

向来电方插播语音:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <visitor id="15" />
  <voicefile>welcome</voicefile>
</Transfer>
```

向去电方插播语音:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <outer id="22" />
  <voicefile>welcome</voicefile>
</Transfer>
```

注:

- 1) 来电转分机, 执行双向插播时, 须先对分机插播再对来电插播, 否则易出现单通问题;
- 2) 模拟分机通过 IP 外线外呼时, 无法实现双向插播, 只能单向插播;
- 3) 插播语音时, 若听到的语音内容不全 (例如前面的几个字未播放), 可以在命令中加一个 silence 文件名, 即 <voicefile>silence+welcome</voicefile>。

参数说明:

参数名称	类型	说明
<ext id>	string	分机号码
<visitor id>	int	来电的编号
<outer id>	int	去电的编号
<voicefile>	string	语音文件，若需插播远程语音文件，应先将语音下载到本地再播放， 点击这里 查看实现方法。

响应示例：

```

HTTP/1.0 200 OK
Date: Tue, 27 Jun 2017 02:51:12 GMT
Server:
X-Frame-Options: DENY
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Access-Control-Allow-Headers: accept, Content-Type
Access-Control-Allow-Origin: (null)
Connection: close
Content-Length: 0
<!--包体为空-->

```

2.2.2 队列 (Queue)

该 API 用于将呼叫转接到队列中，从而实现当被叫忙时主叫可以排队等待，直到被叫可能受理该呼叫。

队列类型：

目前只有两个对象支持队列，分别为：

- 1) 分机队列；
- 2) 分机组队列。

2.2.2.1 来电转分机队列

该 API 用于把来电转接到分机的队列，当分机空闲时 OM 会依次将来电分配给分机。即，

- 1) 如果分机空闲，来电会直接被转接给分机；

2) 如果分机忙，则来电在队列中排队等待，并听等待音乐。当分机空闲后，队列最前面的来电被转接给分机。

注：等待音乐需分机配置，若未配置，则系统自动播放忙音（对于 Rev2.1.5.114 之前的版本，排队时听静音）。

请求示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Queue">
  <visitor id="1"/>
  <ext id="200"/>
</Transfer>
```

参数说明：

参数名称	类型	说明
<visitor id>	int	来电的编号
<ext id>	string	分机号

响应示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="TRANSIENT">
  <visitor id="1" from="02167103750" to="02161208234" callid="16385"/>
  <ext id="200"/>
  <state>wait</state>
</Event>
```

注：其中<state>wait</state>为可选字段。当有该字段时说明分机当前忙，来电在分机队列中等待；没有该字段时说明分机当前空闲，来电已转接给分机。

参数说明：

略，可参考查询来电的 API 响应消息中的参数说明。

（友情提示：超链接返回的快捷键为 Alt + ←）

2.2.2.2 来电转分机组队列

该 API 用于将来电呼叫转接给分机组，OM 会按照该分机组的呼叫分配规则将来电分配给组内的某个/某些空闲分机，从而实现来电和分机组内的某个分机建立通话。

遇忙/无应答处理：

- 1) 分机组具有排队功能，如果分机组内无分机可以受理该来电（如：忙线、挂起、免打扰），则该来电会排队等待，且听该分机组配置的呼叫等待音乐，直到有分机空闲可以受理该来电。
- 2) 如果分机组内有空闲分机可以受理该来电，则该分机会先振铃，摘机后和来电建立通话。如果该分机没有及时应答，则来电自动跳转到下一个空闲分机。

请求示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Queue">
  <visitor id="1"/>
  <group id="1"/>
</Transfer>
```

参数说明：

参数名称	类型	说明
<visitor id>	int	来电的编号
<group id>	int	分机组编号

响应示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="TRANSIENT">
  <visitor id="1" from="13012345678" to="02161208234" callid="20481"/>
  <group id="1"/>
</Event>
```

参数说明：

略，可参考查询来电的 API 响应消息中的参数说明。

（友情提示：超链接返回的快捷键为 Alt + ←）

2.2.3 会议（Conference）

该 API 用于发起会议，目前只支持三方会议。

操作步骤：

- 1) 分机 A 正在和 B 通话；
- 2) 分机 A 把原通话呼叫保持；

- 3) 分机 A 向 C 发起新的呼叫，并建立通话；
- 4) 此时，使用该 API 能够实现以分机 A 为主持方建立 A、B、C 的三方会议。

请求示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Conference">
  <ext id="212"/>
</Transfer>
```

参数说明：

参数名称	类型	说明
<ext id>	string	会议发起方

响应示例：

```
HTTP/1.0 200 OK
Date: Wed, 22 Feb 2017 08:04:44 GMT
Server:
X-Frame-Options:DENY
Cache-Control:no-cache
Pragma:no-cache
Expires:-1
Access-Control-Allow-Headers: accept, Content-Type
Access-Control-Allow-Origin: (null)
Connection: close
Content-Length: 0
<!-- 包体为空 -->
```

实现流程：

步骤 1： 分机 212 呼叫分机 213，建立通话。

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <ext id="212"/>
  <ext id="213"/>
</Transfer>
```

步骤 2： 呼叫保持 213，此时 212 听拨号音，213 听等待音。

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Hold">
  <ext id="212"/>
</Control>
```

步骤 3: 212 呼叫 214，建立通话。

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <ext id="212"/>
  <ext id="214"/>
</Transfer>
```

步骤 4: 执行 Conference 命令，212 作为发起方和 213、214 建立三方会议通话。

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Conference">
  <ext id="212"/>
</Transfer>
```

2.3 语音管理命令 (Manage)

语音管理命令 (Manage) 用来管理语音文件，支持查询和删除设备中的语音文件。

注: 上传和下载语音文件的接口暂未开放，敬请期待。

2.3.1 查询语音文件

该 API 用于查询设备中的所有语音文件。

限制条件:

版本: Rev 2.1.5.124 及以上

请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Manage attribute="Query" >
  <voicefile/>
</Manage>
```

参数说明:

参数名称	类型	参数说明	参数值说明
<voicefile>	string	设备中的语音文件	无值

响应示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Status>
<voicefile>user_file1.pcm|user_file2.pcm|fring2.dat|busytone.dat|ringback.dat|jakevoice_en3.pcm|jakevoice_en2.pcm|jakevoice_en.pcm|jakevoice3.pcm|jakevoice2.pcm|jakevoice.pcm|tts_file.dat|welcome1.pcm|tts_file.pcm|gateway.pcm|hangup.pcm|fring2.pcm|NewMorning.pcm|dnd.pcm|account.pcm|bdomestic.pcm|7_tip.pcm|binternal.pcm|dot_tip.pcm|hangup2.pcm|vm_prompt.pcm|reenter.pcm|2_tip.pcm|record.pcm|busytone.pcm|waiting.pcm|tryagain.pcm|blocal.pcm|nonumber.pcm|6_tip.pcm|dialtone_417.pcm|IVR1.pcm|IVR2.pcm|IVR3.pcm|IVR4.pcm|IVR5.pcm|IVR6.pcm|IVR7.pcm|IVR8.pcm|IVR9.pcm|lock.pcm|silence.pcm|callwaiting.pcm|1_tip.pcm|wrong.pFrom[192.168.130.152:80]:
cm|vm_no.pcm|setting.pcm|dialtone.pcm|netmask.pcm|block.pcm|offhour.pcm|cancel.pcm|enable.pcm|webport.pcm|noanswer.dat|... </voicefile>
</Status>
```

2.3.2 删除语音文件

该 API 用于删除设备中的一个或多个语音文件。

限制条件:

版本: Rev 2.1.5.124 及以上

请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Manage attribute="Remove" >
  <voicefile>user_file1.pcm|user_file2.pcm</voicefile>
</Manage>
```

参数说明:

参数名称	类型	参数说明
------	----	------

参数名称	类型	参数说明
<voicefile>	string	要删除的语音文件名称，“ ”为分隔符。

响应示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="OK">
</Event>
```

注: OM 不会对所删除的语音文件是否存在或是否是系统语音文件做判断, 而交由应用服务器控制。

2.4 来电受理 (Accept)

如果中继的 API 功能开关配置为来电应答前控制模式时, 则每当有电话呼叫该中继, OM 就会将用于描述该事件的 API 报告消息——INVITE 事件推送给应用服务器, 并等待应用服务器在规定时间内 (默认 5 秒) 内对该来电进行控制。

应用服务器可以对来电进行如下类型的控制:

- 1) **Accept**: 如果应用服务器希望 OM 继续受理该来电, 则调用 **Accept API** (即当前 API), OM 将会让该来电进入后续处理流程;
- 2) **Clear**: 如果应用服务器希望 OM 拒接该来电 (如来电为黑名单内的号码), 则调用 **Clear API**, OM 将直接挂断该来电;

超时处理机制:

如果应用服务器未在规定时间内对该来电进行控制, 则默认为应用服务器希望 OM 继续受理该来电, 即按 **Accept** 处理。

请求示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Notify attribute="Accept">
  <visitor id="1"/>
</Notify>
```

参数说明:

参数名称	类型	说明
<visitor id>	int	来电的编号

响应示例:

```
HTTP/1.0 200 OK
Date: Wed, 22 Feb 2017 08:04:44 GMT
Server:
X-Frame-Options:DENY
Cache-Control:no-cache
Pragma:no-cache
Expires:-1
Access-Control-Allow-Headers: accept, Content-Type
Access-Control-Allow-Origin: (null)
Connection: close
Content-Length: 0
<!--包体为空-->
```

2.5 事件报告

当 OM 系统启动、配置变更、分机状态变更或呼叫状态变更等发生时，OM 会自动触发相应的事件，若您的应用服务器开启了监听端口，则可以监听到这些事件。

您可以根据相应事件来监控 OM、分机、线路等的一些状态。

2.5.1 系统事件

此类事件报告主要用于对系统状态变化的汇报。当 OM 系统状态发生变化时，OM 向应用服务器推送此类事件报告。

2.5.1.1 系统启动事件(BOOTUP)

当 OM 系统启动完成后，OM 会向应用服务器推送该报告。

报告示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="BOOTUP">
</Event>
```

2.5.1.2 OM 配置变化事件(CONFIG_CHANGE)

当 OM 的 web 页面配置发生变化时，OM 向应用服务器推送该报告，以便于应用服务器及时更新和同步 OM 的相关配置。

报告示例:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<Event attribute="CONFIG_CHANGE">
</Event>
```

2.5.2 分机状态变更事件

当分机的线路状态发生变化时，OM 设备会向应用服务器推送此类事件报告，以便于应用服务器实时监控分机状态。

分机可汇报的线路状态有四种：空闲(IDLE)、忙线(BUSY)、上线(ONLINE)、离线(OFFLINE)。

2.5.2.1 分机忙事件 (BUSY)

当分机由空闲变成忙碌时，OM 设备会向应用服务器推送该报告。

注： IP 分机由空闲状态下摘机不会立即汇报 BUSY 事件，而是要等拨完号发起呼叫时才会汇报。

报告示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="BUSY">
  <ext id="208" />
</Event>
```

2.5.2.2 分机空闲事件 (IDLE)

当分机由忙碌变成空闲时，OM 设备会向应用服务器推送该报告。

报告示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="IDLE">
  <ext id="208" />
</Event>
```

2.5.2.3 分机上线事件 (ONLINE)

当 IP 分机由离线变为上线，或 IP 分机的地址发生变更时，OM 设备向应用服务器推送该事件报告。

注： 只有 IP 分机有该事件，模拟分机没有。

报告示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="ONLINE" RegisterIP="10.129.91.202" >
  <ext id="208" />
</Event>
```

2.5.2.4 分机离线事件 (OFFLINE)

当 IP 分机由上线变为离线时，OM 设备向应用服务器推送该报告。

注：只有 IP 分机会有该事件，模拟分机没有。

可触发 OFFLINE 事件的情况有：

- 1) IP 分机主动注销；
- 2) 断网断电等原因导致 IP 分机未在注册周期内刷新注册。

报告示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="OFFLINE">
  <ext id="208" />
</Event>
```

2.5.3 呼叫状态变更事件

当分机或中继的呼叫状态发生变化时，OM 向应用服务器推送此类事件报告。

2.5.3.1 振铃事件(RING)

当分机振铃时，OM 向应用服务器推送该事件报告。

报告示例：

- 1) 来电转分机，分机振铃：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="RING">
  <ext id="200" />
  <visitor id="22" from="13012345678" to="02161208234" callid="16408" />
</Event>
```

- 2) 分机呼分机，分机振铃：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="RING ">
  <ext id="200" />
  <ext id="208" />
</Event>
```

- 3) 通过 API 实现分机外呼，且呼叫方式为“先呼被叫，被叫回铃再呼主叫”时，主叫分机振铃：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="RING ">
  <ext id="200" />
  <outer id="22" from="200" to="13012345678" trunk="02161208234" callid="16406" />
</Event>
```

- 4) 通过 API 实现分机外呼或分机呼分机时，且呼叫方式为“先呼主叫，主叫摘机后再呼被叫”时，主叫分机振铃：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="RING ">
  <ext id="200" />
  <visitor from="" />
</Event>
```

- 5) menu 呼分机，分机振铃：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="RING">
  <ext id="200" />
  <menu id="1" />
</Event>
```

2.5.3.2 回铃事件(ALERT)

当被叫对主叫（分机/来电）的呼叫回铃时，OM 向应用服务器推送该事件报告。

报告示例：

- 1) 来电转分机，分机回铃：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="ALERT">
  <visitor id="22" from="13012345678" to="02161208234" callid="16406" />
  <ext id="200" />
</Event>
```

2) 分机呼叫外部电话，外部电话回铃：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="ALERT">
  <ext id="200" />
  <outer to="13012345678" />
</Event>
```

3) 分机呼叫分机，被叫分机回铃：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="ALERT">
  <ext id="200" />
  <ext id="208" />
</Event>
```

4) Menu 外呼，外部电话回铃：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="ALERT">
  <outer id="22" from="" to="13012345678" trunk="02161208234" callid="16406" />
</Event>
```

2.5.3.3 呼叫应答事件(ANSWER)

当分机对呼叫应答时，OM 向应用服务器推送该事件报告。

报告示例：

1) 分机呼分机，被叫分机应答：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="ANSWER">
  <ext id="200" />
  <ext id="208" />
</Event>
```

2) 来电转分机，分机应答：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="ANSWER">
```

```

<ext id="200" />
<visitor id="31" from="13012345678" to="02161208234" callid="16417" />
</Event>

```

2.5.3.4 呼叫被应答事件(ANSWERED)

当主叫检查到被叫应答时，OM 向应用服务器推送该事件报告。

报告示例：

- 1) 分机呼外部电话，分机检查到外部电话应答：

```

<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="ANSWERED">
  <outer id="34" from="200" to="13012345678" trunk="02161208234" callid="16418" />
  <ext id="200" />
</Event>

```

- 2) 分机呼分机，主叫分机检查到被叫分机应答：

```

<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="ANSWERED">
  <ext id="200" />
  <ext id="208" />
</Event>

```

2.5.3.5 通话结束事件 (BYE)

当一路通话释放时，OM 向应用服务器推送该事件报告。

报告示例：

- 1) 来电和分机的通话结束，来电挂断

```

<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="BYE">
  <visitor id="42" from="13012345678" to="02161208234" callid="16426" />
  <ext id="200" />
</Event>

```

- 2) 分机和去电的通话结束，分机挂断

类型 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="BYE">
<ext id="200" />
<outer to="13012345678" />
</Event>
```

类型 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="BYE">
<outer id="38" from="200" to="13012345678" trunk="02161208234" callid="16422" />
</Event>
```

3) 来电转去电的通话结束，来电挂断:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="BYE">
<visitor id="50" from="02167103750" to="02161204000" callid="49202" />
<outer id="51" from="200" to="13012345678" trunk="02161208234" callid="49202" />
</Event>
```

4) 双向外呼的通话结束，两个去电分别各有一个 BYE 事件:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="BYE">
<outer id="48" from="" to="13012345678" trunk="28900928" callid="49200" />
</Event>
```

2.5.3.6 呼叫转移事件 (DIVERT)

当呼叫在 OM 内部发生转移/重定向时，OM 向应用服务器推送该事件报告。

报告示例:

1) 来电 13012345678 呼叫分机时，因分机设置了呼叫转移等原因，导致该呼叫被转移:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="DIVERT">
<visitor id="5" from="13012345678" to="204" callid="12293" />
```

```
</Event>
```

2) 分机 208 呼叫其他分机时，因被叫分机设置了呼叫转移等原因，导致该呼叫被转移：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="DIVERT">
  <ext id="208" />
</Event>
```

2.5.3.7 呼叫临时事件 (TRANSIENT)

该事件通常作为转接 (Transfer) 请求的响应消息，特殊情况下，OM 也会推送该事件消息。

什么情况下会推送该事件：

分机通过模拟中继外呼时，OM 因系统原因不能立刻提供 outer id、trunk、callid，所以会等待呼叫从模拟中继送出时推送该事件，用于对去电信息进行补充。

注：IP 中继没有该现象。

报告示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="TRANSIENT">
  <outer id="66" from="208" to="13012345678" trunk="61208111" callid="8258"/>
  <ext id="208"/>
</Event>
```

呼叫失败事件 (FAILED)

该事件通常作为 API 请求的响应消息出现，特殊情况下，OM 也会推送该事件消息。

哪些情况下会推送该事件：

- 1) 利用 OM API 发起呼叫且该呼叫执行过程中失败时，会触发该事件；
- 2) 如果是手动拨号，只在极个别情况下会触发该事件。

报告格式：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="FAILED">
  <called id=" number "..... />
  <err code="number" reason="failed reason" />
</Event>
```

错误码说明:

错误码字段	说明
<err code="1" reason="temp failure" />	失败
<err code="2" reason="restricted" />	主叫的呼叫权限受限
<err code="3" reason="offline " />	被叫分机不在线
<err code="4" reason="other call" />	被叫分机当前正在和其他终端通话
<err code="5" reason="no circuit" />	中继线资源不足, 无法执行外呼
<err code="6" reason="busy" />	被叫分机忙线
<err code="7" reason="not exist" />	分机不存在
<err code="8" reason="offline" />	主叫分机自身不在线

报告示例:

1) IP 分机离线:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="FAILED">
  <ext id="208" />
  <err code="8" reason="offline" />
</Event>
```

2) 分机 208 正在进行其它通话:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="FAILED">
  <ext id="200" />
  <ext id="208" />
  <err code="4" reason="other call" />
</Event>
```

2.5.4 来电呼入控制流程事件

只有当 API 来电控制开关开启且有来电时, 才会推送该事件。

来电呼入 OM 的处理流程:

- 1) 来电呼叫 OM 中继;
- 2) OM 判断“来电应答前控制”开关是否开启;
- 3) OM 判断“来电应答后控制”开关是否开启;
- 4) 后续处理流程。

来电应答前控制开关:

- 1) 开启状态下, 当来电呼叫相应的外线(中继)时, OM 不会立刻受理该来电, 而是将描述该现象的事件 (INVITE 事件) 推送给应用服务器, 由应用服务器决定是否要受理该来电。如果应用服务器希望 OM 受理该来电, 则执行 **Accept API** 请求; 如果希望 OM 拒绝该来电, 则执行 **Clear API** 请求。如果因网络或系统故障等原因应用服务器未收到 INVITE 事件或收到后未执行控制, 则 OM 等待超时 (该时间由参数 **API_RESPONSE_TO** 控制, 默认为 5s, 设置范围 1-120s, 建议不超过 30s) 后会主动受理该来电。
- 2) 关闭状态下, 当有来电呼叫该外线(中继)时, OM 自动受理该来电, 不会向应用服务器推送 INVITE 事件。

来电应答后控制:

- 1) 开启状态下, 当有来电呼叫外线(中继)时, OM 直接受理该来电, 将其接通并保持静音, 同时将描述该现象的事件 (INCOMING 事件) 推送给应用服务器, 由应用服务器决定如何对该来电进行转接。应用服务器可以执行 **Transfer API** 请求, 将来电转接到语音菜单、分机或分机组等。如果因网络或系统故障等原因应用服务器未收到 INCOMING 事件或收到后未执行控制, 则 OM 等待超时 (该时间由参数 **API_RESPONSE_TO** 控制, 默认为 5s, 设置范围 1-120s, 建议不超过 30s) 后自动将来电转给总机

温馨提示: 为防止应用服务器宕机, Rev2.1.5.112 以后的版本做了修改, 支持超时转总机机制, 对于之前的版本, OM 等待应用服务器处理超时后仍不会处理该来电, 来电方会一直处于静音状态, 直到电话挂断。

- 2) 关闭状态下, OM 会安排自身配置对来电进行转接或导航, 如: 播放欢迎词, 不会推送 INCOMING 事件。

2.5.4.1 来电呼叫请求事件(INVITE)

在中继的“来电应答前控制”开关打开的情况下，当来电呼叫该中继时，OM 会向应用服务器推送 INVITE 事件。

报告示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="INVITE">
  <trunk id="02161204000" />
  <visitor id="31" from="13012345678" to="02161204000" callid="36895" />
</Event>
```

2.5.4.2 来电呼入事件(INCOMING)

在中继的“来电应答后控制”开关打开的情况下，当来电呼叫该中继时，OM 会在应答该来电后，向应用服务器推送 INCOMING 事件。

报告示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="INCOMING">
  <trunk id="02161204000" />
  <visitor id="31" from="13012345678" to="02161204000" callid="36895" />
</Event>
```

2.5.5 按键信息事件(DTMF)

该事件用于汇报用户输入的按键信息。

呼叫（来电/去电/分机）转接到语音菜单（menu）后，用户按键且按键信息满足 menu 的配置条件时，OM 向应用服务器推送该事件。

说明：通过 OM web 页面配置的多级语音导航（IVR）按键时不会上报 DTMF 事件，而功能键*66、*67 的绑定解绑按键会上报 DTMF 事件。

报告示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="DTMF">
  <visitor id="4" from="13012345678" to="02161208234" callid="36868">
    <info>12345</info>
    <menu id="1" />
  </visitor>
</Event>
```

报告包体参数:

参数名称	参数说明
<visitor outer>	来去电
<id>	来去电的编号, 可依据该参数进行呼叫转接
<from>	主叫号码
<to>	被叫号码
<callid>	该路通话的相对唯一的编号
[trunk]	本路呼叫通过的中继线号码
<info>	用户按键信息
[menu]	来去电呼叫驻留的语音菜单

更多示例:

情况 1: 来电接通到语音菜单中, 用户拨号长度达到该菜单的拨号检测长度时, OM 就会向应用服务器推送该报告。(菜单 1 的拨号检测长度为 5, 按键检查结束符为 #, 用户输入 12345)

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="DTMF">
  <visitor id="4" from="13012345678" to="02161208234" callid="36868">
    <info>12345</info>
    <menu id="1" />
  </visitor>
</Event>
```

情况 2: 来电接通到语音菜单中, 用户输入了该菜单的按键检查结束符时, OM 就会向应用服务器推送该报告(菜单 1 的拨号检测长度为 5, 按键检查结束符为 #, 用户输入 200#)。

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="DTMF">
  <visitor id="4" from="13012345678" to="02161208234" callid="36868">
    <info>200#</info>
    <menu id="1" />
  </visitor>
</Event>
```

情况 3: 去电接通到语音菜单中, 用户拨号长度达到该菜单的拨号检测长度时, OM 就会向应用服务器推送该报告。(菜单 1 的拨号检测长度为 5, 按键检查结束符为 #, 用户输入 12345)

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="DTMF">
  <outer id="4" from="200" to="13012345678" callid="68671" trunk="02161208234">
    <info>12345</info>
    <menu id="1" />
  </outer>
</Event>
```

2.5.6 语音文件播放完毕事件(EndOfAnn)

呼叫转接到语音菜单 (menu) 后, OM 播放 menu 的语音文件(voicefile), 当语音文件按照指定播放次数播放完毕后, OM 向应用服务器推送该事件。

注: 语音播放过程中, 若用户按键, 语音停止播放, 且 OM 不推送 EndOfAnn 事件。

报告示例:

来电转 menu:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="EndOfAnn">
  <visitor id="67" from="13012345678" to="208" callid="32835" />
  <menu id="1" />
</Event>
```

去电转 menu:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="EndOfAnn">
  <outer id="4" from="200" to="13012345678" callid="68671" trunk="02161208234" />
  <menu id="1" />
</Event>
```

分机转 menu:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="EndOfAnn">
  <ext id="200" />
  <menu id="1" />
</Event>
```

```
</Event>
```

报告包体参数:

参数名称	参数说明
<visitor > outer>	来去电
<id>	来去电的编号, 可依据该参数进行呼叫转接
<from>	主叫号码
<to>	被叫号码
<callid>	该路通话的相对唯一的编号
[trunk]	本路呼叫通过的中继线号码
<info>	用户按键信息 (暂时无用)
<menu id>	语音菜单

2.5.7 分机组队列事件 (QUEUE)

通过设备页面配置分机组, 且组内分机全忙时, 若有分机或来电呼入分机组, OM 会向应用服务器汇报该事件。

限制条件:

版本要求: Rev2.1.5.118 及以上;

注: 只有通过设备页面配置的分机组进入队列时才会有该事件, 通过 API 配置的分机组没有该事件。

报告示例:

- 1) 组内分机全忙, 分机和来电相继呼入分机组

分机呼入分机组:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="QUEUE">
  <ext id="200" />
  <waiting group="6" count="1" />
</Event>
```

来电呼入分机组:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="QUEUE">
  <visitor id="77" from="18201008888" to="888" callid="45135" />
  <waiting group="6" count="2" />
</Event>
```

2) 若所有分机离线, 分机和来电呼叫分机组

分机呼叫分机组:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="QUEUE">
  <ext id="200" />
  <waiting group="7" count="0" reason="offline" />
</Event>
```

来电呼叫分机组:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="QUEUE">
  <visitor id="77" from="18201008888" to="888" callid="45135" />
  <waiting group="7" count="0" reason="offline" />
</Event>
```

3) 若排队满了, 分机和来电呼叫分机组

分机呼叫分机组:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="QUEUE">
  <ext id="200" />
  <waiting group="7" count="30" reason="full" />
</Event>
```

来电呼叫分机组:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="QUEUE">
  <visitor id="2" />
```

```
<waiting group="7" count="30" reason="full" />
</Event>
```

参数说明:

参数名称	类型	参数说明
<ext id>	string	分机号码, 纯数字非空字符串
<waiting group>	int	分机组的唯一标识, 不同型号设备起始值不同, 最多能添加 50 个分机组 (注意跟分机组的编号区分)
<count>	int	在队列中排队等待的个数, 取值范围 1~30, 超过 30 个后不再排队自动挂断
<visitor id>	int	来电的编号, 用于查询、转接、挂断等操作的判断依据
<from>	string	来电时, 原始主叫号码
<to>	string	来电时, 原始被叫号码 (对于 visitor 而言, 原始被叫为来电呼入的中继号码)
<reason>	string	offline: 表示组内分机全离线; full: 表示组内分机全忙

注:

- 1) 排队等待的电话个数最多为 30 个, 超过 30 个后来电自动挂断, 不再排队;
- 2) Waiting group 的值为分机组的相对唯一标识符, 注意与分机组的组号区分;
- 3) OM 只提供进入队列的事件 (包括当前队列长度), 出队列需 API 应用服务器自己维护。如何判断出队列, 请点击[这里](#)查看方法。

2.6 通话记录报告

(友情提示: 超链接返回的快捷键为 Alt + ←)

通话记录报告简称话单, 是指 OM 对一路通话从开始到结束的记录和统计的报告。当一路通话释放后, OM 向应用服务器实时推送该报告。

CDR 目前一共有 6 种类型: 呼入 IN、呼出 OU、呼叫转移入 FI、呼叫转移出 FW、内部呼叫 LO、双向外呼 CB, 分别对应不同的应用场景。

话单生成情况说明:

IN 话单说明:

- 1) 来电呼入的通话结束后，产生 IN 话单；
- 2) 分机呼分机，通话结束后，被叫分机产生 IN 话单；（主叫产出 LO 话单）

LO 话单说明：

- 1) 分机呼分机的通话结束后，主叫分机产生 LO 话单；
- 2) 来电通过模拟中继呼入转分机，和分机的通话结束后，模拟中继产生 LO 话单；（如无需要可忽略该 CDR）

OU 话单说明：

外呼的通话结束后，产生 OU 话单。

FI 话单说明：

当分机将其呼叫转移给其他号码（该号码为分机或外部电话）时，通话结束后产生 FI 话单。

FW 话单说明：

当分机将其呼叫转移给其他号码（该号码为外部电话）时，通话结束后，产生 FW 话单。

CB 话单说明：

当双向外呼，通话结束时，产生 CB 话单。

报告消息格式：

```
<?xml version="1.0" encoding="utf-8" ?>
<Cdr id="value">
  <callid> value </callid>
  <visitor|outer id=" value " />
  <TimeStart> value </TimeStart>
  <Type> value </Type>
  <Route> value </Route>
  <CPN> value </CPN>
  <CDPN> value </CDPN>
  <TimeEnd> value </TimeEnd>
  <Duration> value </Duration>
  <TrunkNumber> value </TrunkNumber>
  <Recording> value </Recording>
  <RecCodec>PCMU</RecCodec>
</Cdr>
```

参数说明:

参数名称	类型	说明
<Cdr id>	string	话单 id
<callid>	int	通话的相对唯一标识符
[visitor outer id]	int	来电的编号 或者去电的编号，内部分机通话记录无此字段。
<type>	string	IN (打入) /OU (打出) /FI (呼叫转移入) /FW (呼叫转移出) /LO (内部通话) /CB (双向外呼)
<Route>	string	IP (IP 中继) /XO (模拟中继) /IC (内部) /OP (总机)
<TimeStart>	string	呼叫起始时间，即发送或收到呼叫请求的时间
<TimeEnd>	string	呼叫结束时间，即通话的一方挂断的时间
<CPN>	string	主叫号码
<CDPN>	string	被叫号码
<Duration>	int	通话时长，值为 0 说明未接通。
<TrunkNumber>	string	该路通话所经过的中继号码
[Recording]	string	录音文件的相对保存路径，格式：年月日/录音文件 录音文件的命名格式： 1.主叫号码_被叫号码_年月日-时分秒_通话的相对唯一标识符.wav 2.主叫号码_被叫号码_年月日-时分秒_通话的相对唯一标识符 cd.wav 3.主叫号码_被叫号码_年月日-时分秒_通话的相对唯一标识符_cg.wav cg 为主叫录音，cd 为被叫录音。 点击查看录音配置和使用说明
[RecCodec]	string	编码方式，决定录音文件格式，值为：G729、G711(PCMA、PCMU)。 注：本地录音时，G711 编码会生成 pcm 格式录音，设备自动合成 wav 文件存放，可以直接下载播放；G729

参数名称	类型	说明
		编码生成的录音设备不会合成，只保留 send 和 recv 方向的 dat 文件，不能直接播放，需要下载后本地合成，合成方法请点击 这里 查看。

2.6.1 呼入(IN)

来电通过 IP 中继呼入时，会产生 2 个 IN 话单。

- 1) 分机作为被叫的 IN 话单：外部电话 18201008888 呼叫分机 200，通话时长为 2 秒，录音为分机 200 的录音。

```
<?xml version="1.0" encoding="utf-8" ?>
<Cdr id="2820170309162713-0">
  <callid>32818</callid>
  <visitor id="48" />
  <TimeStart>20170309162708</TimeStart>
  <Type>IN</Type>
  <Route>IP</Route>
  <CPN>18201008888</CPN>
  <CDPN>200</CDPN>
  <TimeEnd>20170309162713</TimeEnd>
  <Duration>2</Duration>
  <TrunkNumber>888</TrunkNumber>

  <Recording>20170309/18201008888_200_20170309_162711_8032_cd.wav</Recording>
  <RecCodec>G729</RecCodec>
</Cdr>
```

- 2) 外线作为被叫的 IN 话单：外部电话 18201008888 呼叫外线 888，通话时长为 17 秒，录音为外线 888 的录音（注：需开启外线 888 的监控或控制开关才会收到此话单）。

```
<?xml version="1.0" encoding="utf-8" ?>
<Cdr id="2920170309162714-0">
  <callid>32816</callid>
  <visitor id="48" />
  <TimeStart>20170309162657</TimeStart>
```

```

<Type>IN</Type>
<Route>IP</Route>
<CPN>18201008888</CPN>
<CDPN>888</CDPN>
<TimeEnd>20170309162714</TimeEnd>
<Duration>17</Duration>
<TrunkNumber>888</TrunkNumber>

<Recording>20170309/18201008888_888_20170309_162657_8030_cd.wav</Recording>
  <RecCodec>G729</RecCodec>
</Cdr>

```

来电通过模拟中继呼入时，会产生 2 个话单，分别为 **IN** 和 **LO**（**LO** 话单可过滤掉）。

- 1) 分机作为被叫的 **IN** 话单：外部电话 18201008888 呼叫分机 200，通话时长为 4 秒，录音为分机 200 的录音。

```

<?xml version="1.0" encoding="utf-8" ?>
<Cdr id="3120170309162952-0">
  <callid>32820</callid>
  <visitor id="51" />
  <TimeStart>20170309162945</TimeStart>
  <Type>IN</Type>
  <Route>XO</Route>
  <CPN>18201008888</CPN>
  <CDPN>200</CDPN>
  <TimeEnd>20170309162952</TimeEnd>
  <Duration>4</Duration>
  <TrunkNumber>02161205555</TrunkNumber>

  <Recording>20170309/18201008888_200_20170309_162948_8034_cd.wav</Recording>
  <RecCodec>PCMU</RecCodec>
</Cdr>

```

- 2) 外线作为被叫的 **LO** 话单：外部电话 18201008888 呼叫分机 200，通话时长为 6 秒，录音为外线 02161205555 的录音。

```

<?xml version="1.0" encoding="utf-8" ?>
<Cdr id="5820170309175319-0">
  <callid>32861</callid>
  <visitor id="93" />
  <TimeStart>20170309175311</TimeStart>
  <Type>LO</Type>
  <Route>IC</Route>
  <CPN>18201008888</CPN>
  <CDPN>200</CDPN>
  <TimeEnd>20170309175319</TimeEnd>
  <Duration>6</Duration>
  <TrunkNumber></TrunkNumber>
  <Recording>20170309/18201008888_02161205555_20170309_175313_805D_cg.wav</Recording>
  <RecCodec>PCMU</RecCodec>
</Cdr>

```

2.6.2 呼出(OU)

外呼时，只产生一个 **OU** 话单。

- 1) 分机作为主叫的 **CDR**：分机 200 呼叫外呼电话 18201008888，通话时长为 12 秒，录音为分机的录音。

```

<?xml version="1.0" encoding="utf-8" ?>
<Cdr id="2020170309161117-0">
  <callid>32801</callid>
  <outer id="34" />
  <TimeStart>20170309161101</TimeStart>
  <Type>OU</Type>
  <Route>XO</Route>
  <CPN>200</CPN>
  <CDPN>18201008888</CDPN>
  <TimeEnd>20170309161117</TimeEnd>
  <Duration>12</Duration>
  <TrunkNumber>02161205555</TrunkNumber>
  <Recording>20170309/200_18201008888_20170309_161105_8021_cg.wav</Recording>
  <RecCodec>PCMU</RecCodec>

```

```
</Cdr>
```

- 2) **menu 作为主叫的 CDR:** menu 呼叫外部电话 18201008888, 通话时长为 4 秒, 录音为外线的录音。

```
<?xml version="1.0" encoding="utf-8" ?>
<Cdr id="2520170309162028-0">
  <callid>32809</callid>
  <outer id="41" />
  <TimeStart>20170309162021</TimeStart>
  <Type>OU</Type>
  <Route>IP</Route>
  <CPN></CPN>
  <CDPN>18201008888</CDPN>
  <TimeEnd>20170309162028</TimeEnd>
  <Duration>4</Duration>
  <TrunkNumber>888</TrunkNumber>

  <Recording>20170309/888_18201008888_20170309_162024_8029_cg.wav</Recording>
  <RecCodec>PCMU</RecCodec>
</Cdr>
```

2.6.3 内部互拨(LO)

分机呼分机时, 会产生 2 个话单, 分别为 **LO** 和 **IN** (**IN** 话单可过滤掉)。

- 1) 分机 **200** 作为主叫的 **LO** 话单: 分机 200 呼叫分机 212, 通话时长为 13 秒, 录音为分机 200 的录音。

```
<?xml version="1.0" encoding="utf-8" ?>
<Cdr id="5720170309175055-0">
  <callid>32858</callid>
  <TimeStart>20170309175037</TimeStart>
  <Type>LO</Type>
  <Route>IC</Route>
  <CPN>200</CPN>
  <CDPN>212</CDPN>
  <TimeEnd>20170309175055</TimeEnd>
```

```

<Duration>13</Duration>
<TrunkNumber></TrunkNumber>
<Recording>20170309/200_212_20170309_175042_805A_cg.wav</Recording>
<RecCodec>PCMU</RecCodec>
</Cdr>

```

- 2) 分机 212 作为被叫的 IN 话单：分机 200 呼叫分机 212，通话时长为 4 秒，录音为分机 212 的录音。

```

<?xml version="1.0" encoding="utf-8" ?>
<Cdr id="2220170309161649-0">
  <callid>32806</callid>
  <TimeStart>20170309161641</TimeStart>
  <Type>IN</Type>
  <Route>IC</Route>
  <CPN>200</CPN>
  <CDPN>212</CDPN>
  <TimeEnd>20170309161649</TimeEnd>
  <Duration>4</Duration>
  <TrunkNumber></TrunkNumber>
  <Recording>20170309/200_212_20170309_161644_8026_cd.wav</Recording>
  <RecCodec>PCMU</RecCodec>
</Cdr>

```

2.6.4 呼叫转移(FI/FW)

呼叫转移号码为内部分机号码时，只产生 FI 话单。

分机作为被叫的 FI 话单：外部电话 02167103750 呼叫分机 200，但分机 200 将该呼叫转移到分机 201 了，且 02167103750 和 201 的通话时长为 4 秒。

```

<?xml version="1.0" encoding="utf-8" ?>
<Cdr id="5820821018163856-0">
  <callid>90</callid>
  <visitor id="89" />
  <TimeStart>20821018163827</TimeStart>
  <Type>FI</Type>
  <Route>IC</Route>
  <CPN>02167103750</CPN>
  <CDPN>200</CDPN>

```

```

<TimeEnd>20821018163856</TimeEnd>
<Duration>4</Duration>
<TrunkNumber>02161208234</TrunkNumber>
</Cdr>

```

呼叫转移号码为外部电话或手机号码时，会产生两个话单，分别为 **FI** 和 **FW** 话单。

- 1) 分机作为被叫的 **FI** 话单：外部电话 02167103750 呼叫分机 200，但分机 200 将该呼叫转移到 13012345678 了，且 02167103750 和 13012345678 的通话时长为 4 秒。

```

<?xml version="1.0" encoding="utf-8" ?>
<Cdr id="5820821018163856-0">
  <callid>90</callid>
  <visitor id="89" />
  <TimeStart>20821018163827</TimeStart>
  <Type>FI</Type>
  <Route>IC</Route>
  <CPN>02167103750</CPN>
  <CDPN>200</CDPN>
  <TimeEnd>20821018163856</TimeEnd>
  <Duration>4</Duration>
  <TrunkNumber>02161208234</TrunkNumber>
</Cdr>

```

- 2) 分机作为被叫的 **FW** 话单：外部电话 02167103750 呼叫分机 200，但分机 200 将该呼叫转移到 13012345678 了，且 02167103750 和 13012345678 的通话时长为 4 秒。

```

<?xml version="1.0" encoding="utf-8" ?>
<Cdr id="5920821018163856-0">
  <callid>90</callid>
  <visitor id="89" />
  <TimeStart>20821018163827</TimeStart>
  <Type>FW</Type>
  <Route>XO</Route>
  <CPN>200</CPN>
  <CDPN>13012345678</CDPN>
  <TimeEnd>20821018163856</TimeEnd>
  <Duration>4</Duration>

```

```
<TrunkNumber></TrunkNumber>
</Cdr>
```

2.6.5 双向外呼(CB)

双向外呼时，产生两个 **CB** 话单。

- 1) 外部电话作为主叫的 **CB** 话单：外部电话 13012345678 呼叫外部电话 02167103750，通话时长为 9 秒。

```
<?xml version="1.0" encoding="utf-8" ?>
<Cdr id="420170328135454-0">
  <MAC>00:0E:A9:3D:00:0C</MAC>
  <callid>20484</callid>
  <outer id="4" />
  <TimeStart>20170328135442</TimeStart>
  <Type>CB</Type>
  <Route>IP</Route>
  <CPN>13012345678</CPN>
  <CDPN>02167103750</CDPN>
  <TimeEnd>20170328135454</TimeEnd>
  <Duration>9</Duration>
  <TrunkNumber>50015</TrunkNumber>
  <Recording>20170328/50015_02167103750_20170328_135445_5004_cg.wav</Recording>
  <RecCodec>PCMU</RecCodec>
</Cdr>
```

- 2) 外部电话作为被叫的 **CB** 话单：外部电话 13012345678 呼叫外部电话 02167103750，通话时长为 6 秒。

```
<?xml version="1.0" encoding="utf-8" ?>
<Cdr id="520170328135455-0">
  <MAC>00:0E:A9:3D:00:0C</MAC>
  <callid>20484</callid>
  <outer id="5" />
  <TimeStart>20170328135445</TimeStart>
  <Type>CB</Type>
  <Route>IP</Route>
```

```
<CPN></CPN>
<CDPN>02167103750</CDPN>
<TimeEnd>20170328135455</TimeEnd>
<Duration>6</Duration>
<TrunkNumber>50016</TrunkNumber>
<Recording>20170328/50016_02167103750_20170328_135449_5005_cg.wav</Recording>
<RecCodec>PCMU</RecCodec>
</Cdr>
```

3 实现方案

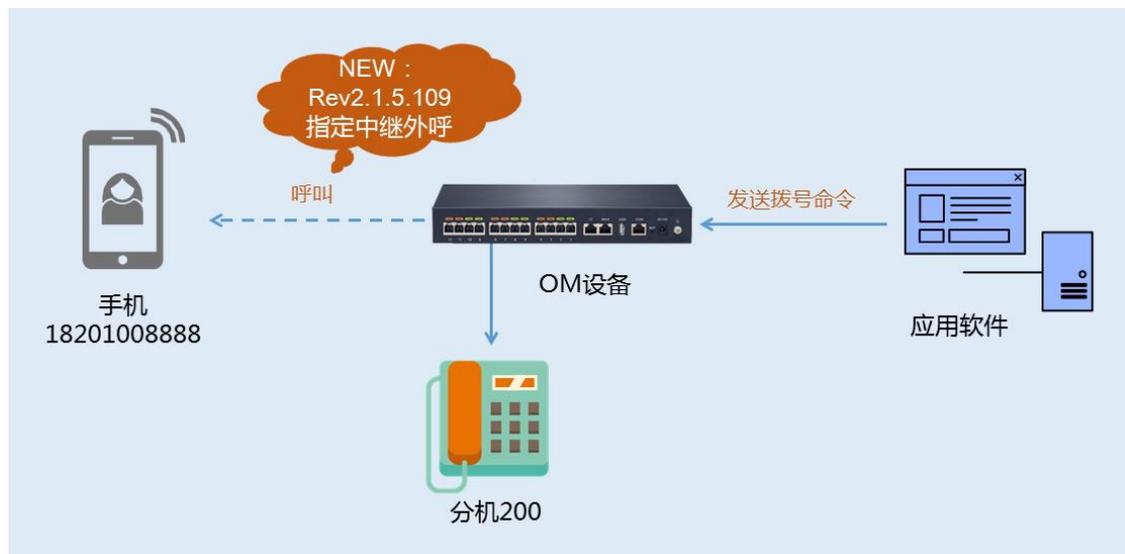
本章介绍一些常见的功能实现方案，详情请看下文。

3.1 点击拨号

点击拨号是呼叫中心、CRM 系统等应用的一个常见功能，用户只要在软件/Web 页面上点击某个联系人系统就会自动向该联系人发起呼叫，方便高效。

3.1.1 实现原理

原理图如下所示：



注：应用服务器和应用客户端即开发者自行研发的软件，如：呼叫中心、CRM 系统、办公软件等。

3.1.2 步骤描述

- 1) 用户在应用客户端/web 页面上对某个联系人执行点击拨号，客户端向应用服务器发送拨号请求；
- 2) 应用服务器判断用户要拨打的号码属于内部分机还是外部电话（如：手机号）。若是分机，则调用分机呼分机的命令；若是外部电话，则调用分机外呼的命令。当然此时的主叫分机号码（ext id）为用户的本机号码；
- 3) 应用服务器通过 HTTP 协议将相应的 API 命令发送给 OM，OM 收到消息后会先进行权限认证，认证通过后执行该命令；
- 4) OM 执行该拨号命令，按照一定的顺序分别呼叫主叫和被叫，双方都应答后通话建立。

3.1.3 OM API 点击拨号的特点

支持两种呼叫方式:

- 1) 效率高型——先呼被叫，被叫回铃时再呼主叫；
- 2) 体验好型 ——先呼主叫，主叫摘机后，再呼被叫。

支持 SIP 实体话机作为主叫时自动应答功能。

呼叫命令:

常用的点击拨号命令分为“分机呼分机”和“分机外呼”两种。

- 1) 分机呼分机 API 命令:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <ext id="200"/>
  <ext id="208"/>
</Transfer>
```

- 2) 分机外呼 API 命令:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <ext id="208"/>
  <outer to="13012345678"/>
</Transfer>
```

3.2 来/去电弹屏

名称: 来去电弹屏。

定义: 当坐席的分机收到某个客户的来电或向某个客户发起去电时，其电脑上自动弹出该客户的详细资料。

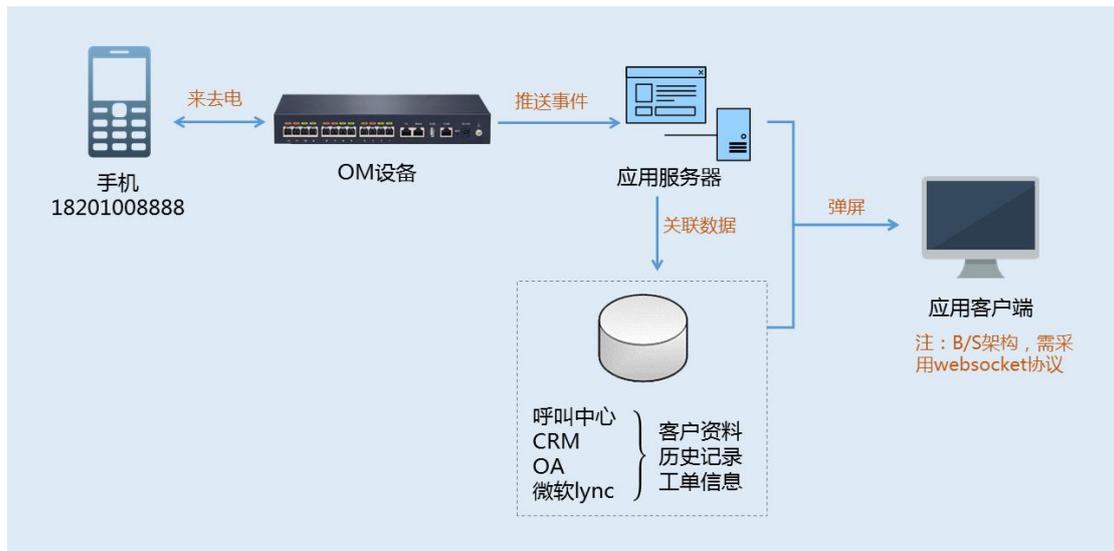
目的: 提高坐席的工作效率和服务质量。

执行时间: 来去电的第一时间。

3.2.1 实现原理

- 1) 分机在完成一次呼叫的过程中，其呼叫状态会不断变化，如：振铃、接通、挂断等，OM 会将这些事件消息实时推送给开发者开发的应用服务器；
- 2) 应用服务器收到这些消息后，再控制该分机绑定的应用终端执行来去电弹屏、开始通话计时，结束计时等动作。

原理图：



应用服务器收到哪个消息时执行弹屏：

分机的呼叫状态变化事件有 8 个，分别为：RING、ALERT、ANSWER、ANSWERED、BYE、DIVERT、TRANSIENT、FAILED。跟弹屏相关的事件有：RING、ALERT、TRANSIENT。

3.2.2 来电弹屏

当收到包含 ext 和 visitor 的 RING 事件时，执行来电弹屏。

消息示例：

手机 13012345678 呼叫分机 200，分机 200 开始振铃：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="RING">
  <ext id="200" />
  <visitor id="22" from="13012345678" to="02161208234" callid="16408" />
</Event>
```

3.2.3 去电弹屏

当收到包含 `outer` 和 `ext` 的 `ALERT` 和 `TRANSIENT` 事件时，执行去电弹屏。

- 1) `TRANSIENT` 事件是针对模拟中继的；
- 2) `ALERT` 事件主要是针对 `SIP` 中继的。

说明：一般情况下，电信营业商给用户提供的模拟线路默认都是没有反极性信号的，如果用户申请了反极性信号，则收到 `TRANSIENT` 事件就无须执行弹屏了，只有当收到 `ALERT` 事件时才执行去电弹屏，具体原因不在此展开说明。

消息示例：

- 1) 分机 200 呼叫外部电话 13012345678，OM 将该呼叫从模拟外线送出：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="TRANSIENT">
  <outer id="2" from="200" to="13012345678" trunk="306" callid="16386" />
  <ext id="200" />
</Event>
```

- 2) 分机 200 呼叫外部电话 13012345678，OM 检测到外部电话的回铃信息（即，外部电话开始振铃）：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="ALERT">
  <outer id="2" from="200" to="13012345678" trunk="306" callid="16386" />
  <ext id="200" />
</Event>
```

3.2.4 B/S 架构下如何实现来去电弹屏

应用服务器收到 API 事件后，需将事件转发给浏览器，告知页面弹屏，此时由于 HTTP 协议不支持服务器主动访问浏览器，所以需采用另外一种协议——WebSocket 协议。



3.3 多级 IVR

多级 IVR 是呼叫中心的常见应用。

OM API 实现多级 IVR 要用到语音菜单(menu)这个对象。

3.3.1 配置要求

开启外线的来电应答后控制开关和状态监控开关；

说明：来电应答后控制，即来电呼入 OM 并接通后，来电转 IVR 还是人工完全由应用服务器托管控制实现。

3.3.2 相关 API 列举

事件：

- 1) INCOMING 事件：中继(也称为外线)的来电应答后控制开启的情况下，当来电通过该中继呼入时，OM 首先会应答该来电，并向应用服务器推送 INCOMING 事件，然后等待应用服务器通过 API 进行呼叫控制；
- 2) DTMF 事件：呼叫转接到 menu 后，当用户按键满足 menu 的按键信息汇报条件时，OM 向应用服务器推送该事件；
- 3) EndOfAnn 事件：呼叫转接到 menu 后，menu 的语音文件按照指定规则播放完毕后，OM 向应用服务器推送该事件。

请求：

- 1) Assign 请求：配置语音菜单 (menu)，用于之后的使用。配置项包括：播放的语音文件、播放次数、按键检测长度、判断输入结束字符；
- 2) Connect 请求：用于各个菜单之间的跳转和转接到人工，如来电转语音菜单、来电转分机。

3.3.3 简易的 IVR 实现流程图



3.3.4 异常处理

如果用户没有按照 IVR 要求输入，应用服务器需要对这些异常情况进行处理。

异常现象：

- 1) 判断输入超时：当应用服务器收到的 DTMF 事件中 info 值以字母 T 结束，说明用户在该菜单下输入超时；
- 2) 判断输入错误：当应用服务器收到的 DTMF 事件中 info 值不在该菜单定义的有效值范围内，说明用户输入错误。

举例：

- 1) 首位输入超时的 DTMF 事件：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="DTMF">
  <visitor id="4" from="13012345678" to="02161208234" callid="36868">
    <info>T</info>
    <menu id="1" />
  </visitor>
</Event>
```

- 2) 输入一半超时的 DTMF 事件：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="DTMF">
  <visitor id="4" from="13012345678" to="02161208234" callid="36868">
    <info>1234T</info>
    <menu id="1" />
  </visitor>
</Event>
```

异常处理方法：

当遇到这些情况时，一般需要给用户播放纠错提示音，播放完后继续转接回原来的 menu。

不过，现在有种更简便的方法，直接转回原来的 menu，同时临时播放纠错提示音。

这里重点介绍转接的一个扩展功能——**临时替换语音文件**。

格式如：

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <visitor id="41" />
  <menu id="10"/>
  <voicefile>input_wrong+input_id_card_no</voicefile>
</Transfer>
```

临时替换语音文件说明：

- 1) 如果呼叫转接 menu 时临时指定新的 voicefile, 则 OM 不再向通话方播放 menu 原配置的 voicefile, 而播放该临时指定的 voicefile, 且仅播放一遍, 播放完毕汇报 EndOfAnn 事件;
- 2) DTMF 的汇报规则依然遵循 menu 的配置(即, infolength 和 exit 两个参数依然按 menu 的配置生效);
- 3) 临时指定的 voicefile 仅在本次转接生效。

举例：

比如, menu 10 的配置为:

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
  <menu id="10">
    <voicefile>input_id_card_no</voicefile>
    <repeat>1</repeat>
    <infolength>20</infolength>
    <exit>#</exit>
  </menu>
</Control>
```

其中, 语音文件 input_id_card_no 的内容为: “请输入您的身份证号, 按井号键结束”。

- 1) 当来电转接到该 menu 后, 用户听到提示音“请输入您的身份证号, 按井号键结束”, 然后用户输入了“123#”, 应用服务器收到 DTMF 事件:

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="DTMF">
  <visitor id="41" from="13012345678" to="02161208234" callid="36868">
    <info>123#</info>
    <menu id="1" />
  </visitor>
</Event>
```

- 2) 当应用服务器判断出用户输入的数据有误, 依然将来电转接到 menu 10, 并同时指定 voicefile 为 input_wrong+input_id_card_no 两个文件。(其中, 语音文件 input_wrong 的内容为“您的输入有误”, +号用来连接多个文件)。

这样，来电还是在 menu 10，但听到提示音变为“您的输入有误，请输入您的身份证号，按并号键结束”。

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <visitor id="41" />
  <menu id="10"/>
  <voicefile>input_wrong+input_id_card_no</voicefile>
</Transfer>
```

3) 然后，用户输入了正确的身份证号“410410199001010101#”，应用服务器收到：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="DTMF">
  <visitor id="41" from="13012345678" to="02161208234" callid="36868">
    <info>410410199001010101#</info>
    <menu id="1" />
  </visitor>
</Event>
```

如此，即完成一次异常处理。

3.4 来电黑名单

实现来电黑名单需设备配合开启外线的“来电应答前控制”和“状态监控”开关。

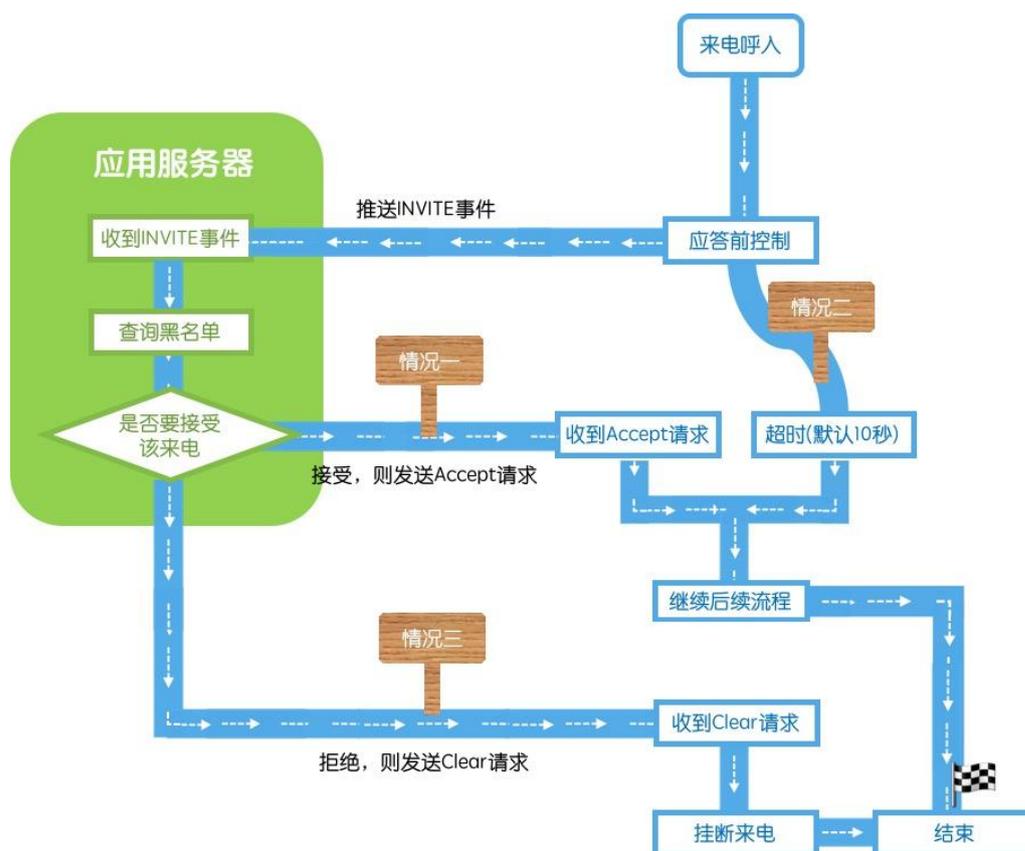
来电应答前控制开关说明：

- 1) 开启状态下，当来电通过该外线呼入时，OM 会向应用服务器推送 INVITE 事件，并等待应用服务器进行控制。应用服务器如果要接受该来电则执行 **Accept** 请求，如果要拒接该来电则执行 **Clear** 请求，如果应用服务器一直未做处理，则等待超时（默认 5 秒）后 OM 会自动接受该来电；
- 2) 关闭状态下，OM 会默认接受该来电。

配置如下图所示：



来电黑名单的实现流程图：



用到的 API:

- 1) INVITE 事件(来电呼叫请求事件);
- 2) Accept 请求(接受来电);
- 3) Clear 请求(拒接来电)。

3.5 来电直通车

3.5.1 痛点

如果客户拨打您的企业电话时，每次都要转接很多次之后才能找到要找的人，那么这个客户很有可能因不耐烦而流失掉。

同样，如果您的员工给客户打电话，但客户未接听到，而当客户回拨过来后却根本找不到是谁打给他的，那这个客户也很可能流失掉了。

其实，您与客户的距离并不遥远，而是差了一个来电直通车。

3.5.2 简介

来电直通车，即将客户号码和坐席的分机或手机绑定，当客户来电时直接转给对应的坐席。来电直通车可以通过 OM API 轻松实现。

3.5.3 配置要求

开启外线的来电应答前控制和状态监控开关。

3.5.4 实现流程

- 1) 当 OM 收到来电时，向应用服务器推送 INVITE 事件。

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="INVITE">
  <trunk id="02161204000" />
  <visitor id="31" from="13012345678" to="02161204000" callid="36895" />
</Event>
```

其中，visitor from 字段为来电的电话号码，visitor id 字段用于做操作标识，trunk id 为外线号码。

- 2) 应用服务器收到 INVITE 事件后，根据数据库记录的绑定关系，判断该来电号码绑定了哪个坐席。

- 3) 如果来电号码没有绑定坐席，则执行 **Accept** 请求让 OM 受理该来电，接下来走正常的处理流程。

```
<?xml version="1.0" encoding="utf-8" ?>
<Notify attribute="Accept">
<visitor id="31"/>
</Notify>
```

其中，visitor id 即之前收到的 INVITE 事件中的 visitor id。

注：如果来电号码绑定了坐席，同样执行 **Accept** 请求让 OM 受理该来电，但延时 1 秒后将来电转给绑定的坐席。

步骤 1:受理该来电。

```
<?xml version="1.0" encoding="utf-8" ?>
<Notify attribute="Accept">
<visitor id="31"/>
</Notify>
```

步骤 2:为了给 OM 时间接通该来电，所以延时 1 秒。

步骤 3:等来电接通后，将来电转给绑定的坐席。

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <visitor id="31"/>
  <ext id="200"/>
</Transfer>
```

其中，ext id 是绑定坐席的分机号。

3.6 指定中继外呼

指定中继外呼的方法有两种，分别为加前缀外呼和指定 trunk id 外呼。

3.6.1 加前缀外呼

在呼叫号码之前加前缀，外呼时随机选择任意一条空闲的模拟/IP 外线呼出。

外呼条件：

设备需配置加前缀的外呼规则：

登录设备，进入**基本设置 > 拨号规则 > 外呼规则**，选择外呼拨号方式为“外线加前缀，内线直播”，配置前缀和选线方式。如下图所示：



命令格式：

如，OM 设备外呼规则为，外呼前缀为 9 全部走 SIP 外线，被叫号码为 13012345678，命令格式为：

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <ext id="208"/>
  <outer to="9,13012345678"/>
</Transfer>
```

3.6.2 指定 trunk id 外呼

版本要求：Rev2.1.5.109 及以上。

命令格式：

在外呼的 API 接口命令中增加<trunk id>节点，则外呼时从某一固定中继呼出。

如，指定中继 888，命令格式为：

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <ext id="208"/>
```

```

    <trunk id="888" />
    <outer to="13012345678"/>
</Transfer>

```

参数说明:

参数名称	类型	说明
<ext id>	string	分机号码
<outer to>	string	呼出的被叫号码
[trunk id]	string	中继号码, 可以为 IP 中继或模拟中继

支持此类指定中继外呼的场景包括: 分机呼外部电话、来电转外部电话、去电转外部电话、语音菜单呼外部电话。

特殊情况:

回拨(双向外呼)指定中继外呼时不支持上文中的指定 trunk id 形式, 其指定中继时需借助 display 字段, 格式如下:

```

<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
    <outer to="13012345678" display="50018" />
    <outer to="02167103750" display="50019" />
</Transfer>

```

参数说明:

参数名称	类型	说明
<outer to>	string	呼出的外部电话号码, 第一个为主叫方, 第二个为被叫方
<display>	string	指定中继号码, 可以为 IP 中继或者模拟中继号码, 需要打开 API_DISPLAY 和 API_CPN 参数

3.7 酒店叫醒解决方案

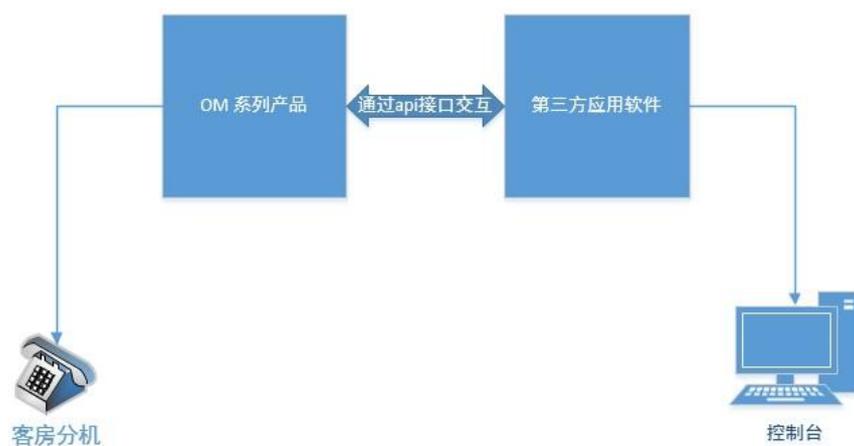
3.7.1 场景描述

- 1) 向酒店服务台申请叫醒服务, 并提供房号和叫醒时间等信息;
- 2) 酒店服务人员将以上信息录入到“酒店叫醒系统”中;

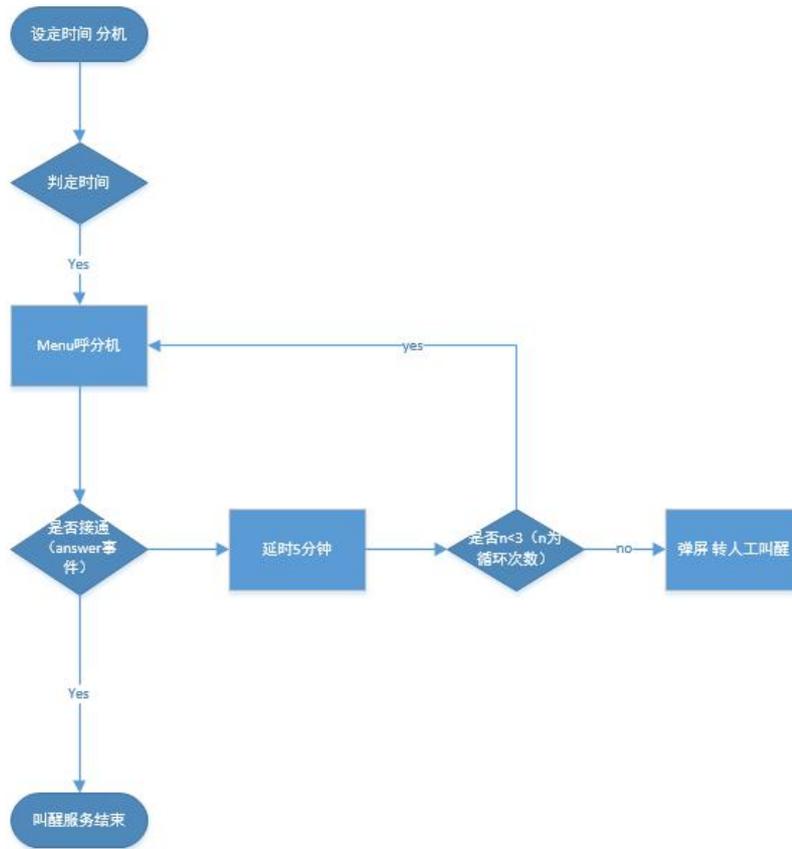
- 3) 当客人要求的叫醒时间到达时,“酒店叫醒系统”控制“电话交换机”自动呼叫客人房间的话机;
- 4) 客人房间的话机振铃, 客人接听电话, 即完成自动叫醒的目的;
- 5) 若客房话机自动叫醒两次后仍无人接听, 则酒店叫醒系统弹窗, 提醒工作人员执行人工叫醒。

3.7.2 实现方案

实现原理图如下:



实现流程:



流程描述:

- 1) 通过 PC 控制台设置叫醒时间以及叫醒客房分机号;
- 2) 叫醒应用软件判断时间到后, 通过 api 接口向 OM 发送 xml 文本消息: **menu** (语音菜单) 呼分机, 客房分机振铃;

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<Transfer attribute="Connect">
```

```
  <menu id="1"/>
```

```
  <ext id="212"/>
```

```
</Transfer>
```

- 3) 若客人接听电话, 收到如下所示的 **ANSWER 事件**, 则认为客人已被叫醒; 否则, 应用服务器会延迟 5 分钟后再次发送 menu 呼分机命令:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<Event attribute="ANSWER">
  <ext id="212" />
  <menu id="1"/>
</Event>
```

4) 若发送两次后，仍无人应答，控制台弹屏，提醒工作人员去人工叫醒。

3.8 API+TTS 实现方案

（友情提示：超链接返回的快捷键为 Alt + ←）

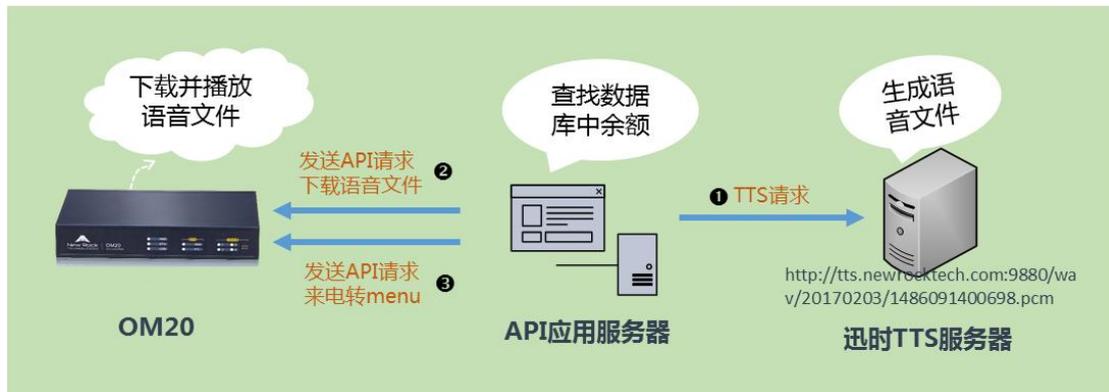
迅时自主开发了一套 TTS 服务器（仅供测试使用，不能商用，商用需开发者自己准备 TTS 服务器）。开发者可利用 TTS 服务器动态合成语音文件，并利用 OM API 实时播放。

实现场景如：客户来电后，按键进入语音导航，选择“1 账户查询”，语音播报“王先生，您的账号余额为 300 元。”

版本要求：

软件版本 Rev 2.1.5.109.18 及其以上（若版本不支持，可联系客服帮您升级 bin 包或直接更新 wget 组件（wget 1.9.2 及以上），客服电话：4007779719 或 4006172700）。

实现方案图：

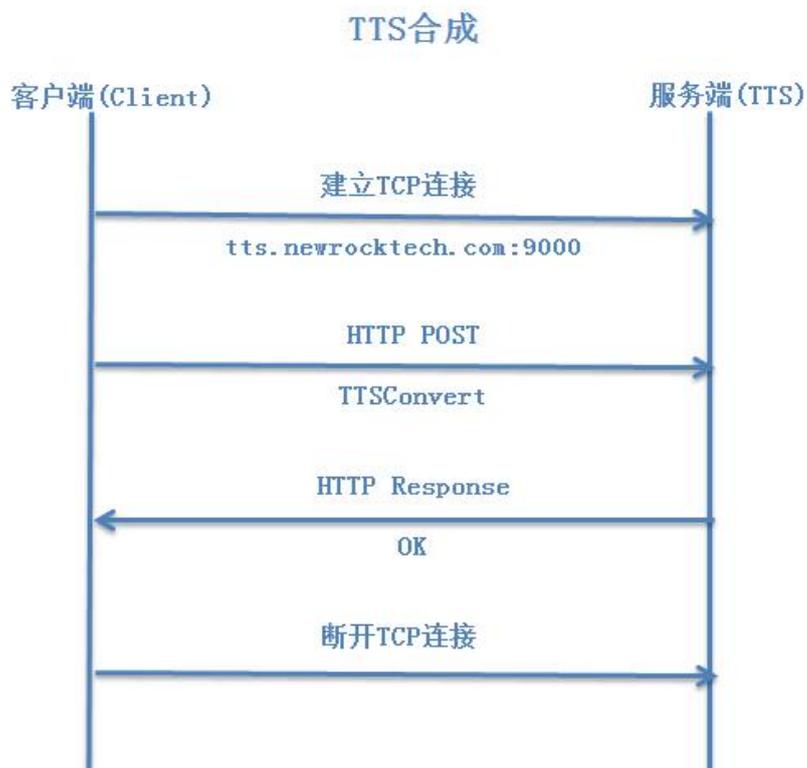


语音合成：

迅时提供 TTS 服务器，可以合成 OM 需要的多种语音格式文件，实际合成时可能只需要一种格式，推荐使用 pcm，设备直接下载不需要转换格式。

[点击这里](#)下载合成 TTS 语音文件的 demo。

TTS 接口通信图:



语音文件下载图:



接口实现:

TTS 请求:

```

POST /xml HTTP/1.1
Host: tts.newrocktech.com:9000
Cache-Control: no-cache
    
```

Content-Type: text/xml

Content-Length: 195

```
<?xml version="1.0" encoding="utf-8"?>
<Control attribute="TTSCConvert">
  <content>王先生，您的账号余额为 300 元</content>
  <lang>ch_female</lang>
  <speed>5</speed>
  <volume>5</volume>
  <wav>0</wav>
  <pcm>1</pcm>
  <dat>0</dat>
</Control >
```

参数说明:

<>表示必选项，[]表示可选项（当参数值为默认值或空时，响应消息可能不携带该参数），| 表示或者关系。

参数名称	子项说明	值说明
<content>	语音内容	支持中文(需 utf8 编码)和英文,最多 130 个汉字及标点
<speed>	语速	1~9
<volume>	音量	1~9
<lang>	发言人	支持 ch_female ch_male en_female, 分别为中文女声 中文男声 英文女声
<wav>	是否需要 wav	0: 不需要, 1: 需要
[pcm]	是否需要 pcm	0: 不需要, 1: 需要
[dat]	是否需要 dat	0: 不需要, 1: 需要

TTS 成功响应:

```
<?xml version="1.0" encoding="utf-8"?>
<Event attribute="OK">
  <wav_url>http://tts.newrocktech.com:9880/wav/20170203/1486091400698.wav</wav_url>
  <pcm_url>http://tts.newrocktech.com:9880/wav/20170203/1486091400698.pcm</pcm_ur
```

```

|>
<dat_url>http://tts.newrocktech.com:9880/wav/20170203/1486091400698.dat</dat_url>
</Event>

```

参数说明:

参数名称	子项说明	值说明
wav_url	wav 文件下载路径	无
pcm_url	pcm 文件下载路径	无
dat_url	dat 文件下载路径	无

TTS 失败返回:

```

<?xml version="1.0" encoding="utf-8"?>
<Event attribute="FAILED">
  <error code="number" reason="failedreason" />
</Event>

```

参数说明:

错误码字段	说明
<error code="400" reason="Bad Request" />	请求错误
<error code="421" reason="XML Error " />	XML 格式错误
<error code="422" reason="xx Param Unrecognized" />	xx 参数无法识别
<error code="423" reason="xx Param Value Error" />	xx 参数值错误
<error code="500" reason="Server Error" />	服务器错误

动态播放合成的语音:

- 1) 首先, 需要临时配置 menu, 指定远程 PCM 路径, 下载 TTS 语音文件;
- 2) 其次, 使用 transfer 指令转接播放。

配置 menu 命令如下:

```

<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">

```

```

    <menu id="1">
      <voicefile>http://tts.newrocktech.com:9880/wav/20170203/1486091400698.pcm</voicefile>
      <repeat>1</repeat>
      <infolength>3</infolength>
      <exit>#</exit>
    </menu>
  </Control>

```

Transfer 转接命令如下:

```

<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <visitor id="41" /> //当有来电时监听到的 visitor id
  <menu id="1"/>
</Transfer>

```

温馨提示:

由于动态播放远程合成语音需要下载, 有可能来不及导致播放失败, 需要注意以下几点:

- 1) 动态合成语音不宜过长, 简短 10~30 字, 控制文件大小;
- 2) 动态语音不宜频繁, 同时配合网络, 保证网络下载畅通;
- 3) menu 配置 id 范围 1~50, 不能重复防止转接互相错乱, 临时播放使用过后的 menu id 可以重用;
- 4) 配置 menu 和转接 tranfer 之间可以等待 100ms~1s 左右, 防止语音文件下载失败;
- 5) TTS 服务器域名需要解析, OM 设备和应用服务器都需要 DNS 服务器配置以解析域名;
- 6) TTS 服务器固定域名: tts.newrocktech.com, 合成端口 9000, 返回下载端口 9880。

3.9 语音播报实现方案

场景需求:

来电呼入, 根据语音导航按键后语音播报“您的账户余额为 2 元 3 角 5 分”。 如何实现?

前提条件:

开启外线的“来电应答后控制+状态监控”开关。

原理介绍:

OM API 支持**数字+语音文件**组合播放。将文字录制成语音文件，与数据库中查询到的数字组合使用即可。

实现原理图：



实现方法：

实现步骤如下：

步骤 1：上传语音文件。

登录 OM 设备，点击**基本设置 > 语音管理 > 上传欢迎词**，上传语音文件（语音文件制作方法：[点击下载](#)并安装风声软件，输入语音文件内容，合成语音）或选择**文字转欢迎词**，如 user_1：“您的账户余额为”，user_2：“元”，user_3：“角”，user_4：“分”。

步骤 2：将来电转接到语音菜单（menu）。

执行来电转 menu 命令，并指定临时播放的语音文件，即在命令中增加<voicefile>节点，如下命令中第五行：

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <visitor id="41" />
  <menu id="1"/>
  <voicefile>user_1+2+user_2+3+user_3+5+user_4</voicefile>
</Transfer>
```

注:

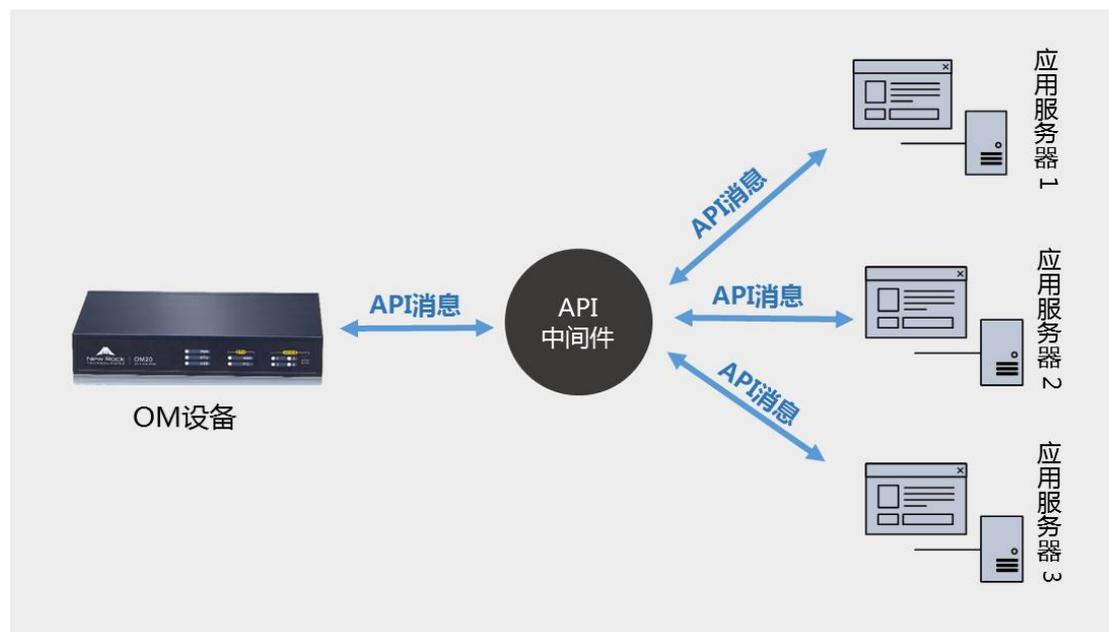
- 1) <voicefile>指定语音文件个数小于等于 10，只要语音文件存在（可下载）就可以正常播放，超过 10 个后不再播放（V113.2 修复：从第十一个语音文件去匹配内部语音文件，如果匹配到，就会播放，找不到，就报错）；
- 2) "数字"为从数据库中实时查询到的当前话费余额值。

3.10 API 中间件解决方案

API 中间件用来解决一个 OM 设备同时向多个应用服务器发送消息的问题。

它的特点是只转发数据，不进行逻辑判断，即只要 OM 向 API 中间件推送事件，API 中间件就会把该事件推送给应用服务器。同时，应用服务器向 API 中间件发送的消息，中间件也会将其转发给 OM。

不过需注意的是，应用服务器可以有多个，而控制服务器只能有一个，否则容易打乱 OM 的程序。



配置使用：

3.10.1 OM 配置

登录 OM 设备，配置认证服务器地址（此处的服务器地址应为 API 中间件所在电脑的 IP 地址，端口号可任意填写，如 8989，建议范围在 1024~65535 内），如下图所示：



3.10.2 API 中间件配置

点击[这里](#)下载 API 中间件安装包。

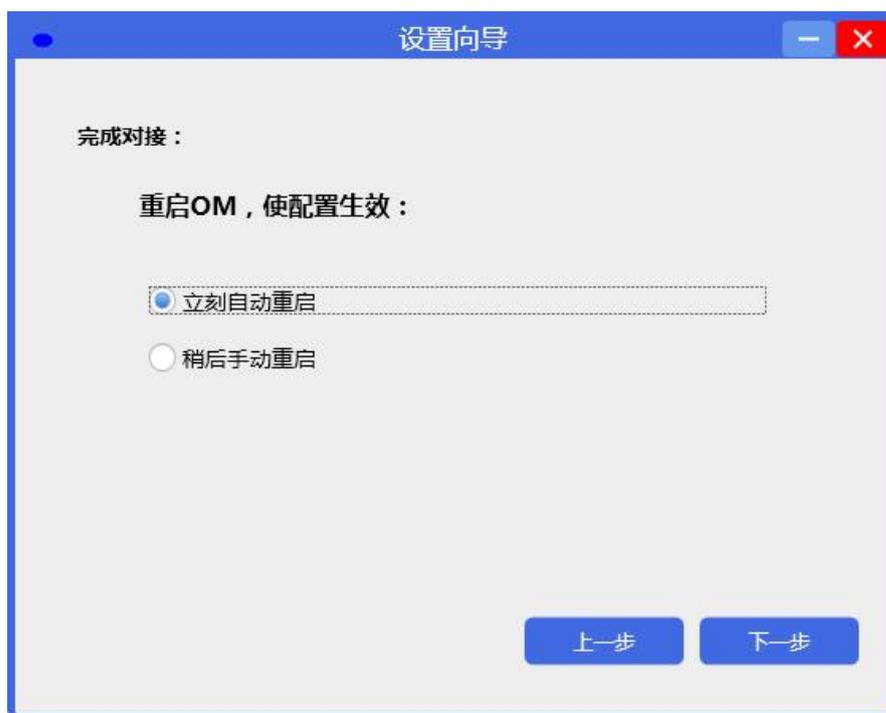
步骤 1：安装并打开 API 中间件，配置 OM 地址和密码，点击“登录”。如下图所示：



步骤 2：登录后会跳转到设置向导，填写监听端口，如 8989（应与在 OM 配置的端口号 8989 保持一致）。



步骤 3：一直点击“下一步”，重启设备。



步骤 4: 完成了基本配置，一直点击“下一步”，然后退出向导，进入如下界面。



步骤 5: 点击红框内“添加”，即可添加多个客户端。

- 1) 服务器地址应填写为你的应用服务器（或监听工具）所在电脑的 IP 地址；
- 2) 端口号应为中间件发送消息的端口号，可自定义，但不可与设备发送消息的端口号 8989 相同，如设置为 5050。



注：API 开关需确保为打开状态。

步骤 6： 点击上图中“关闭”按钮后，出现如下界面。

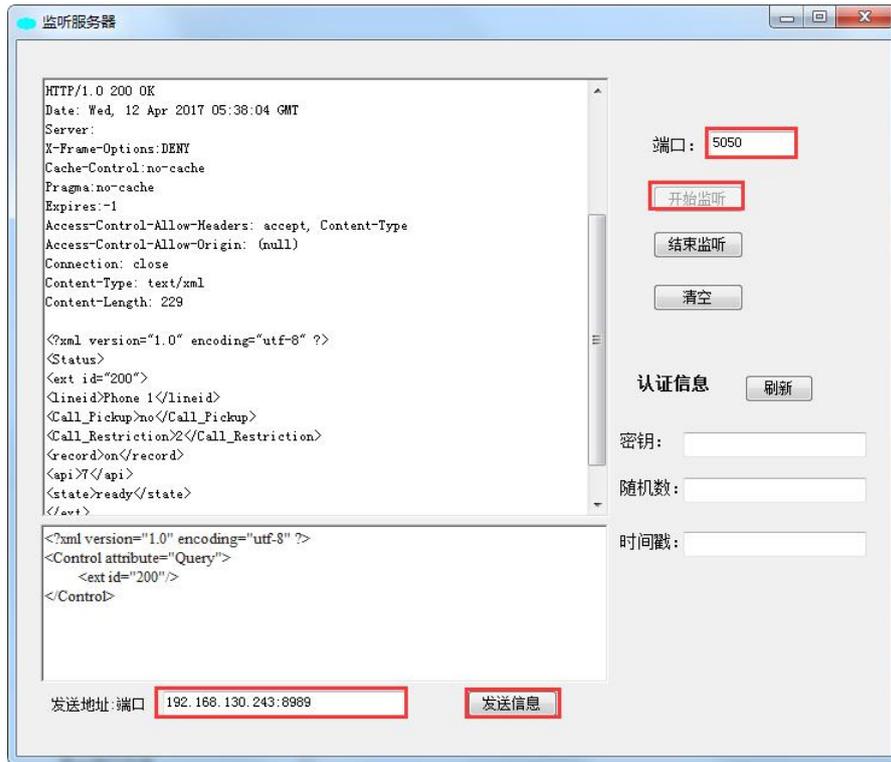


步骤 7： 点击“启用”按钮，即可启用中间件。

3.10.3 应用服务器配置

这里以测试工具为例。

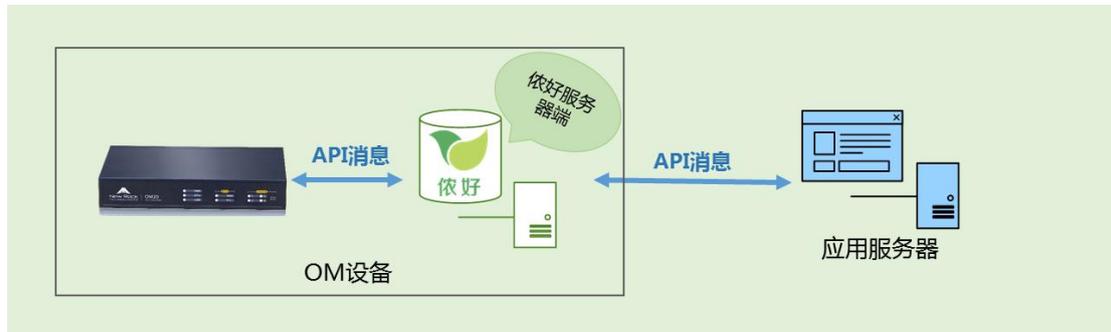
- 1) **消息接收：** 配置监听端口，这里应为 5050（用来接收中间件发送的消息），点击“开始监听”。
- 2) **消息发送：** 配置发送地址和端口，发送地址应为 API 中间件所在服务器的 IP 地址，发送端口应为 API 中间件接收消息的端口，即 8989。



如有任何问题，欢迎加 API 技术交流群：371554591 交流。

3.11 API+依好解决方案

依好控制及转发 API 话单+状态消息，实现原理图如下：



限制条件：

依好服务端版本：V2.3.2 及以上。

配置方法：

OM 设备启用依好，并配置 API 消息转发信息，依好转发话单到第三方应用平台，失败会保存到内置存储，供第三方应用平台下载更新。

依好服务端支持 http 远程配置依好的 API 转发服务器信息。

步骤 1: 登录 OM 设备，进入应用服务器 > API，选择“依好”，保存。



步骤 2: 在 OM 浏览器配置 URL，如：

http://192.168.130.122:8787/Auth?method=gw.config.set&api_server_addr=192.168.130.119:8888&&url=api/call.html&cdr=1&online=1&offline=1&pass=admin。

URL 说明:

路径	/Auth?method=gw.config.set	必须项
参数 api_server_addr	ip:port, 支持域名	可选项
参数 url	api server 的 url	可选项
参数 cdr	话单记录, 1 转发, 0 不转发, 默认 0	可选项
参数 online	IP 分机上线, 1 转发, 0 不转发, 默认 0	可选项
参数 offline	IP 分机离线, 1 转发, 0 不转发, 默认 0	可选项
参数 all	API 所有事件, 1 转发, 0 不转发, 默认 0, 其他全 0 时默认 1	可选项
参数 wait	CDR 转发等待响应 success, 1 等待, 0 不等待, 默认 0, 当设为 1 时等待响应 success, 否则认为失败会缓存话单, 等待超时时间 3 秒	可选项
参数 voicekey	满意度评分按键设置信息, 默认 *88	可选项
参数 savecall	只保留通话记录, 不转发, 与 api_server_addr 不共用	可选项
参数 callday	通话记录保存天数, 默认保存 30 天, 最大一年 365 天	可选项

路径	/Auth?method=gw.config.set	必须项
参数 filemin	通话记录保存文件命名，默认按一小时 60， 可以设置 10，20，30 分钟	可选项
参数 pass	web 管理员密码	必选项

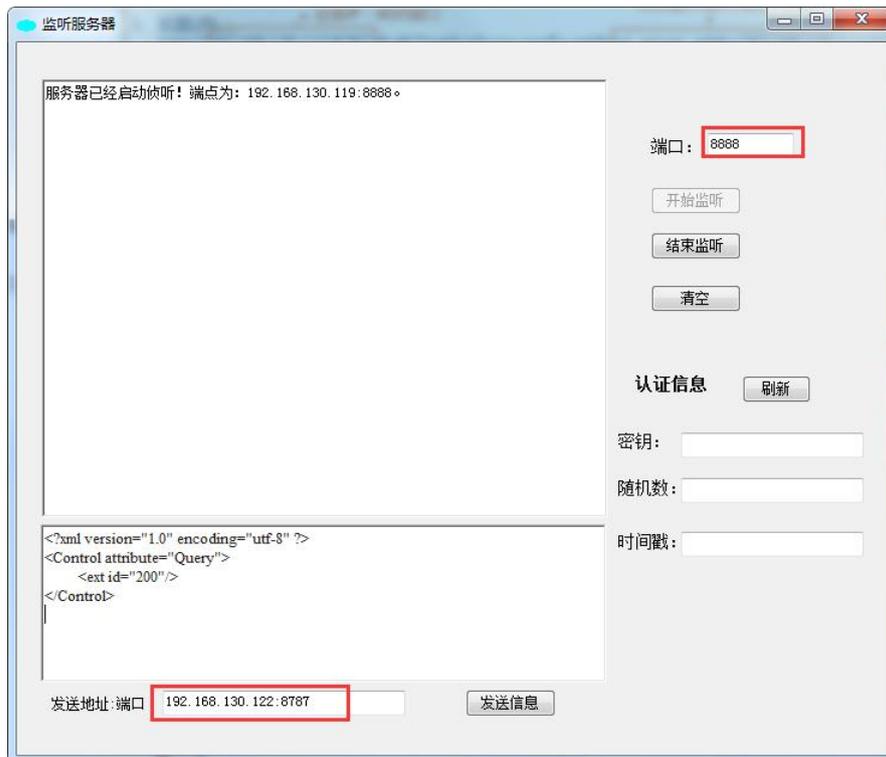
配置返回 ok 成功，即时生效。

应用系统平台只需要配置 **cdr=1**，让依好服务端转发话单，转发失败会重传一次，重传失败保存内置存储的 APICdr 目录，供应用系统下载更新。

- 1) 如果是转发，话单保存格式: APICdr/cdr_yyyymmdd.txt。
- 2) 如果是只保留，话单保存格式: APICdr/yyymmdd/cdr_hh.txt。

测试使用：

下面以测试工具为例，向 OM 发送 API 消息测试下吧。



监听端口：需和通过配 OM URL 中的应用服务器端口保持一致，应为 8888；

- 1) **发送地址和端口：**OM 设备 IP 地址及依好端口，如 192.168.130.122:8787。

注：API 与依好同时启用时，API 只能接收话单和状态消息，不能使用控制功能。

3.12 分机外呼控制实现方案

应用场景：

监狱通话系统，OM 分机拨号外呼，API 实现分机外呼号码检测限制。

环境要求：

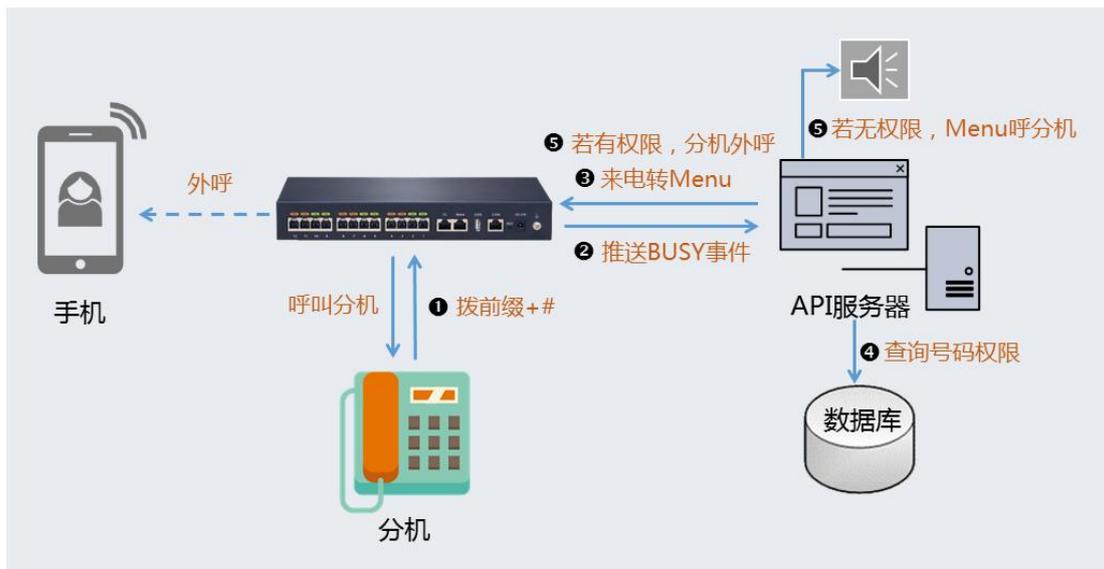
版本要求：Rev 2.1.5.118.5 及以上。

OM 外呼规则：加前缀模式，前缀 8 绑定不可用外线，前缀 9 绑定正常外线。

注：

- 1) API 分机无控制模式，必须使用外呼规则前缀方式实现；
- 2) 分机摘机，必须先拨前缀“8#”，听到语音提示后拨打外呼号码。

实现原理图：



实现流程：

步骤 1：录制语音文件上传 OM，配置 API 语音菜单。

Menu1 提示音“请输入外呼号码，以#号结束”。

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
  <menu id="1">
    <voicefile>welcome</voicefile>
```

```

    <repeat>5</repeat>
    <infolength>20</infolength>
    <exit>#</exit>
  </menu>
</Control>

```

Menu2 提示音“您无权限拨打该号码，请重新输入”。

```

<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
  <menu id="2">
    <voicefile>welcome</voicefile>
    <repeat>5</repeat>
    <infolength>20</infolength>
    <exit>#</exit>
  </menu>
</Control>

```

注：welcome 为系统语音文件名称，可根据需求自行修改。

步骤 2：分机摘机拨打前缀“8#”，OM 会上报分机 BUSY 事件到 API 服务器。

```

<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="BUSY">
  <ext id="200" />
</Event>

```

步骤 3：服务器检测 BUSY 后，使用 menu1 语音菜单，连接分机，播报提示音输入外呼号码。

```

<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <menu id="1"/>
  <ext id="200"/>
</Transfer>

```

步骤 4：分机按键，上报 DTMF 事件，收集外线号码信息。

```

<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="DTMF">

```

```

    <ext id="200" />
        <info>13012345678#</info>
        <menu id="1" />
</Event>

```

步骤 5: API 服务器连接数据库，校验外呼号码。

若符合：执行分机外呼命令，从而建立通话。

```

<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
    <ext id="200"/>
    <outer to="9,13012345678"/>
</Transfer>

```

若不符合：转接 menu2 提示号码无权限，请重新输入号码；可循环转接 menu2，直到输入号码正确或用户挂机。

```

<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
    <menu id="2"/>
    <ext id="200"/>
</Transfer>

```

3.13 监听+插播+强拆实现方案

应用场景：

应用于监狱系统，当犯人与外线建立通话，犯人绑定的狱警话机振铃，摘机进入监听状态，可通过话机按键或者应用服务器发送控制命令实现监听、插播的功能。

环境要求：

版本要求：Rev 2.1.5.119.1 及以上。

权限配置：监听和插播方分机须开启“监听”开关

实现原理图：**实现流程：**

步骤 1： 犯人分机与外部电话通话过程中，执行如下命令进入监听状态。

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Monitor">
  <ext id="201"/> //狱警
  <ext id="200"/> //犯人
</Control>
```

执行命令后，狱警监听犯人和外线的通话，犯人和外线听不到狱警的声音。

步骤 2： 监听过程中若遇到犯人说敏感信息，执行如下命令进入插播状态。

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Talk">
  <ext id="201"/>
</Control>
```

进入插播状态后，外线听保持音，狱警与犯人通话。

步骤 3： 插播结束后，执行 listen 命令，返回到监听的状态。

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Listen">
  <ext id="201"/>
</Control>
```

狱警监听犯人与外线间的通话。

步骤 4： 监听或插播过程中，若应用服务器检测到犯人的电话余额不足，或者已经达到规定的通话时长，可向犯人插播一段语音，提示犯人余额不足，或通话时长已到，请挂机。

```
<?xml version="1.0" encoding="udf-8" ?>
<Transfer attribute="Connect">
```

```
<ext id="200" />
<voicefile>welcome</voicefile>
</Transfer>
```

其中 welcome 为设备系统语言文件，需更换为实际需要的语音文件。

步骤 5: 在通话过程中，若狱警觉得有必要打断通话，可一键强拆。

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Clear">
  <ext id="200"/>
</Control>
```

3.14 接通后播放语音实现方案

应用场景：

来电时，坐席摘机后，对来电方播放语音，如“2202 号话务员为您服务”。

限制条件：

版本要求：Rev 2.1.5.117 及以上。

实现原理图：



实现流程：

步骤 1: 录制语音文件，如 user_service: “号话务员为您服务”，上传至 OM。

步骤 2: API 执行来电转接分机，指定临时播放的语音文件为两个静音文件，命令如下。

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <visitor id="14"/>
  <ext id="200"/>
</Transfer>
```

```
<voicefile>silence+silence</voicefile>
```

```
</Transfer>
```

步骤 3: 收到外线的被应答事件 (ANSWERED) 时, 执行插播语音。

对来电方插播语音的 API:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<Transfer attribute="Connect">
```

```
  <visitor id="15" />
```

```
  <voicefile>2202+user_service</voicefile>
```

```
</Transfer>
```

说明:

- 1) “2202”为分机对应的工号;
- 2) 来电方听语音过程中, 分机静音, 语音播放完毕后, 自动进入通话;
- 3) 来电方听语音的同时, 若想让分机也听一段提示音, 可同时对分机插播语音。

向分机插播语音的 API:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<Transfer attribute="Connect">
```

```
  <ext id="200" />
```

```
  <voicefile>语音文件名</voicefile>
```

```
</Transfer>
```

注:

- 1) [点击这里](#)了解插播语音的详细接口内容。
- 2) 若需要插播的语音文件为远程语音文件, 需先下载到本地再插播, [点击这里](#)查看实现方法。

3.15 满意度评价实施方案

应用场景：

客服与客户通话（来电/去电），当服务结束后，需要把通话转接到一个满意度评价语音系统，提示客户对此次服务进行评价，客户做出评价，系统记录评价结果；若客户超时未做出评价，挂断通话。

实现原理图：



环境要求：

版本要求：Rev117 及以上。

权限配置：配置分机（客服电话）api 参数值为 0xF。

注：

- 1) 使用客户端转满意度调查不需要配置。
- 2) 具体配置方法见文档 6.2。

实现流程：

- 1) 配置语音文件：该语音文件用于转接到满意度评价语音提示。

语音文件 1：非常满意请按 1，基本满意请按 2，不满意请按 3。

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
  <menu id="1">
    <voicefile>evaluate</voicefile>
```

```

        <repeat>3</repeat>
        <infolength>1</infolength>
        <exit>#</exit>
    </menu>
</Control>

```

语音文件 2 : 您已超时, 通话即将挂断。

```

<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
    <menu id="2">
        <voicefile>timeout</voicefile>
        <repeat>1</repeat>
        <infolength>1</infolength>
        <exit>#</exit>
    </menu>
</Control>

```

语音文件 3:感谢您的评价。

```

<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
    <menu id="3">
        <voicefile>thanks</voicefile>
        <repeat>1</repeat>
        <exit>#</exit>
    </menu>
</Control>

```

- 2) 通话中, 客服通过电话机按键, 汇报 DTMF 事件给应用服务器; 若客服处有应用系统, 可跳过该步骤, 在应用服务系统做按键转接满意度评价。

去电的按键事件。

```

<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="DTMF">
    <ext id="300" />

```

```

    <info>5</info>
    <outer id="1" from="300" to="788" trunk="309" callid="32769" />
</Event>

```

来电的按键事件。

```

<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="DTMF">
    <ext id="300" />
    <info>7</info>
    <visitor id="1" from="788" to="309" callid="32772" />
</Event>

```

- 3) 根据按键事件，或者应用服务系统的转接满意度评价请求，应用服务器把呼叫转接语音菜单 1，请求评价。

```

<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
    <visitor/outer id="1" />
    <menu id="1"/>
</Transfer>

```

- 4) 等待接收客户的评价结果，若未接收到客户按键信息（菜单 1 的 DTMF 事件）。

若收到 endofann 事件延时 5s 仍未收到 DTMF 事件，再次执行步骤 3,重复两次后仍未收到客户的按键信息，执行来电转接语音菜单 2。

```

<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
    <visitor/outer id="1" />
    <menu id="2"/>
</Transfer>

```

收到菜单 2 的 endofann 事件，挂断通话。

```

<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Clear">
    <visitor/outer id="1"/>

```

```
</Control>
```

5) 等待客户的按键信息，接收到客户的按键信息（菜单 1 的 DTMF 事件）。

记录满意度评价的结果，info 信息。

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="DTMF">
  <visitor id="3" from="788" to="309" callid="32771">
    <info>1</info>
    <menu id="1" />
  </visitor>
</Event>
```

接收到按键事件后，把通话转接到语音菜单 3，语音提示感谢评价。

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <visitor/outer id="1" />
  <menu id="3"/>
</Transfer>
```

收到语音菜单 3 的 EndOfAnn 事件，把通话挂断。

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Clear">
  <visitor/outer id="1"/>
</Control>
```

3.16 公网服务器访问私网内 OM

针对服务器部署在公网，OM 部署在私网的场景，我们提供如下三种解决方案，您可以根据实际情况选择其中一种。

- 1) 端口映射；
- 2) 客户端直接访问 OM。

二种方案的区别：

端口映射：

优点：点对点，直接采用 IP 认证方式即可；

缺点：不适合网络环境较复杂的场景。

客户端直接访问 OM：

优点：解决跨域访问问题后，客户端可直接访问 OM。

缺点：目前只有 OM20、OM50 支持，且要求 OM 版本 Rev 2.1.5.100 及以上。

注：此方案要求采用数字签名认证方式。

3.16.1 端口映射

此方案，需要解决两个问题：NAT 问题和 API 认证问题。

部署场景：

解决 NAT 问题，可通过以下两种常见的部署场景：

场景 1：OM 直接做动态域名配置



此场景非常简单，只需一步配置即可：

登录 OM，点击“基本设置>外网访问”，在外网地址处选择“内置 DDNS”，并填写动态域名账号、密码和域名（需要自行申请，如花生壳）。



注意:

当采用“动态域名”这种方式时，数据的源 IP 地址在转发过程中会被修改掉，而由于 OM API 默认认证方式是对数据的源 IP 地址进行认证，所以会出现认证失败的现象。故，此时须采用数字签名认证方式。

场景 2: OM 端口映射到私网路由器，私网路由器做动态域名配置。



配置方法:

设备以 OM20 为例，路由器以 TP-LINK 为例，型号 TL-WVR450G。

步骤 1: 查看 OM20 的 SIP 端口、RTP 端口和 API 端口。

查看方法如下:

- 1) SIP 端口 (UDP 协议): 5060

进入 **外线设置 > SIP 外线 > 注册选项**



2) RTP 端口 (UDP 协议) : 10010-10266。



3) API 端口 (HTTP 协议) : 80。

API 端口和 web 端口一致，默认为 80。

步骤 2: 路由器配置端口映射。

1) 进入系统服务 > 动态 DNS > 花生壳动态域名，填写用户名、密码和域名。



2) 进入传输控制 > NAT 设置 > 虚拟服务器, 填写 SIP、API、RTP 端口映射, 映射完成可在服务列表中看到状态“已启用”(SIP 和 RTP 的服务协议类型是 UDP, API 是 TCP)。



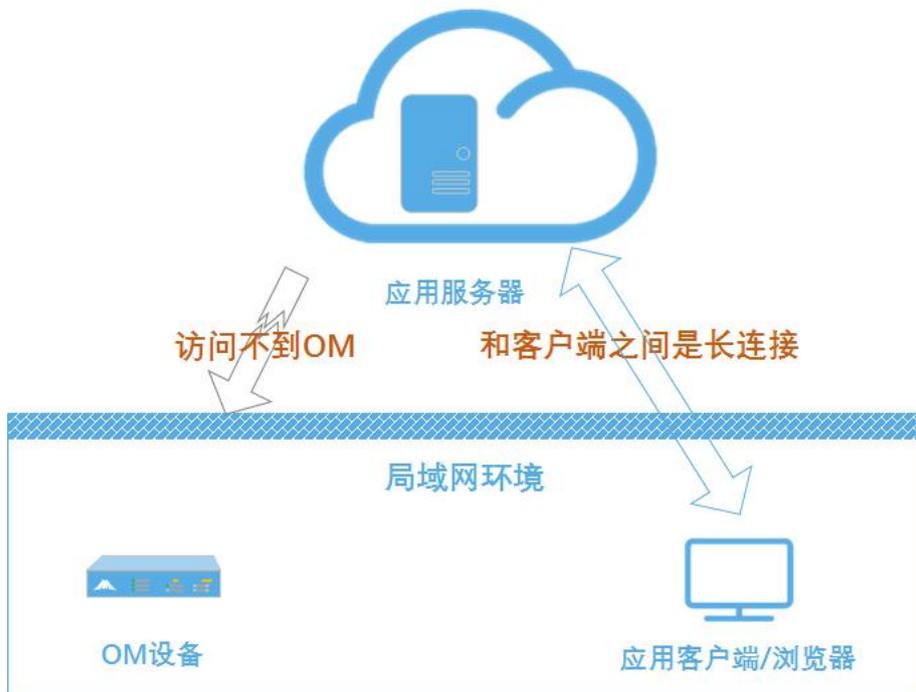
步骤 3: 动态域名配置。

登录 OM，点击“**基本设置 > 外网访问**”，在外网地址处选择“**外部主机名（入口路由器 DDNS）**”，并填写已申请好的动态域名。

**3.16.2 客户端直接访问 OM****应用场景:**

当应用服务器部署在公网，而 OM 设备部署在局域网，同时坐席终端（客户端/浏览器）和 OM 设备又在同一个局域网内。

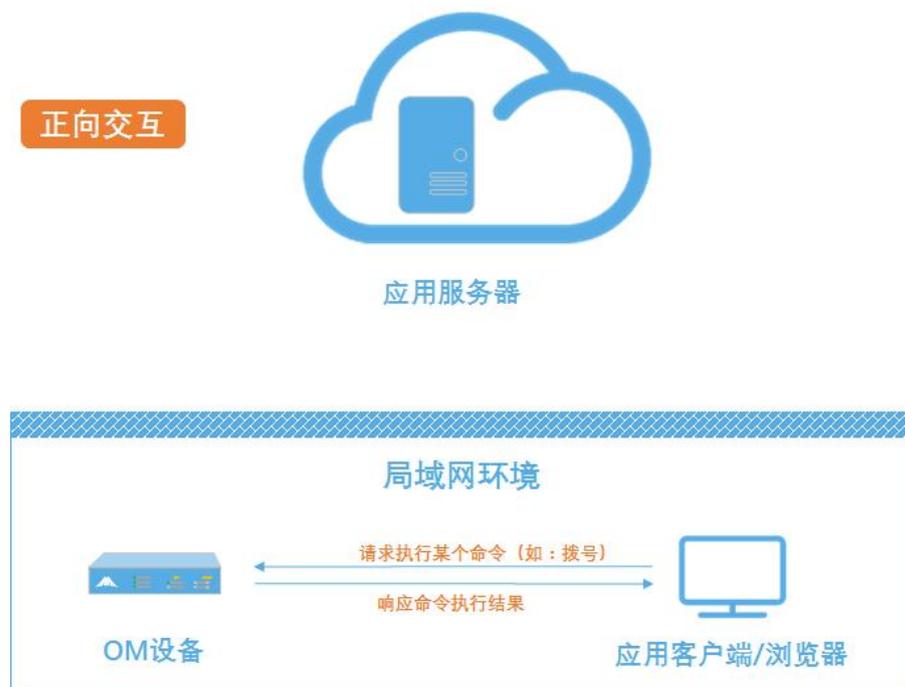
此时，应用服务器是无法访问到 OM 的。



解决方案:

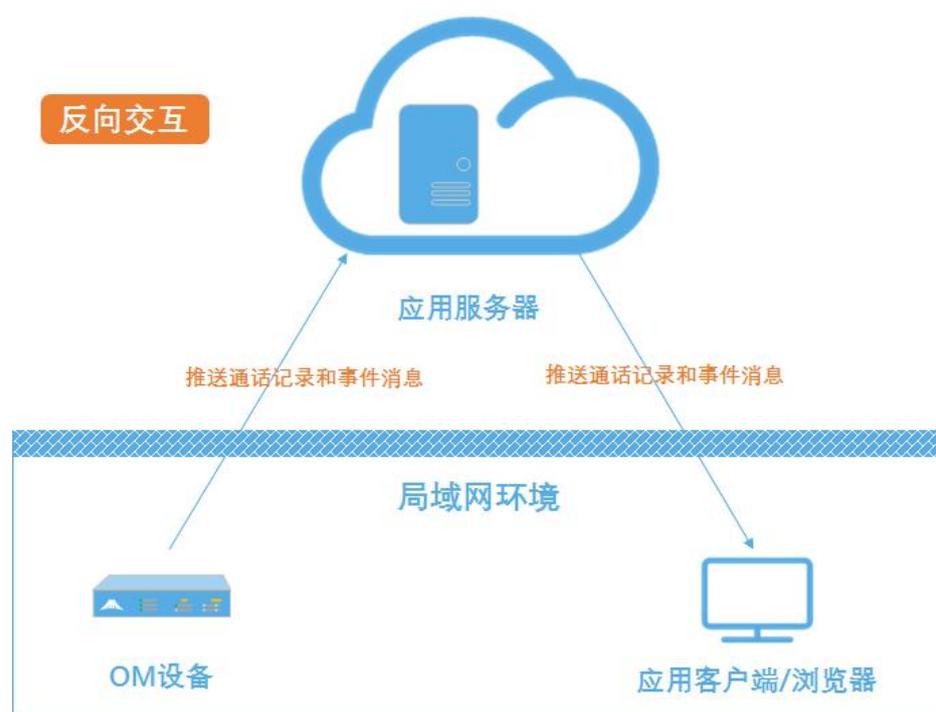
1) 正向交互:

让应用客户端/浏览器直接向 OM 发送 API 请求, 而非通过应用服务器。



2) 反向交互:

当 OM 触发事件或生成通话记录时, 推送给应用服务器, 再由应用服务器转发给客户端/浏览器。



API 应用服务器向 OM 发送消息时，须解决两个问题：

- 1) 浏览器的跨域问题：（参考资料：[\[如何解决跨域访问 OM 的问题\]](#) (/chapter4/projectIntro/domain.md)）；
- 2) API 认证问题。（参考资料：[OM API 数字签名认证原理介绍](#)）。

4 专题介绍

本章给大家介绍一些重难点知识、概念。

4.1 IPPBX

(友情提示: 超链接返回的快捷键为 Alt + ←)

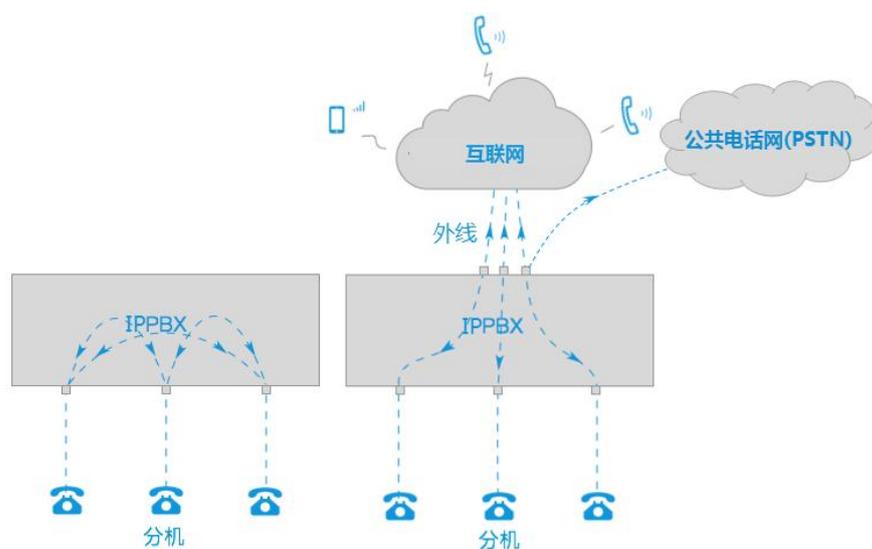
IPPBX 是基于 IP 技术的 PBX (电话交换系统), 可以通过互联网进行电话交换和通信, 而且支持更加丰富的通信内容, 除了语音, 还有视频、文本等。

这种通过互联网进行语音通信的技术被称为 VOIP (voice over ip)。

OM 就是迅时出品的 IPPBX 设备, 主要致力于解决中小企业的融合通信问题。

主要有两个作用:

- 1) 提供多个内部分机 (包括模拟分机和 IP 分机), 且分机之间可以免费通话。
- 2) 连接公共电话网和互联网, 使得内部分机和外部的固话或手机之间可以相互通话。

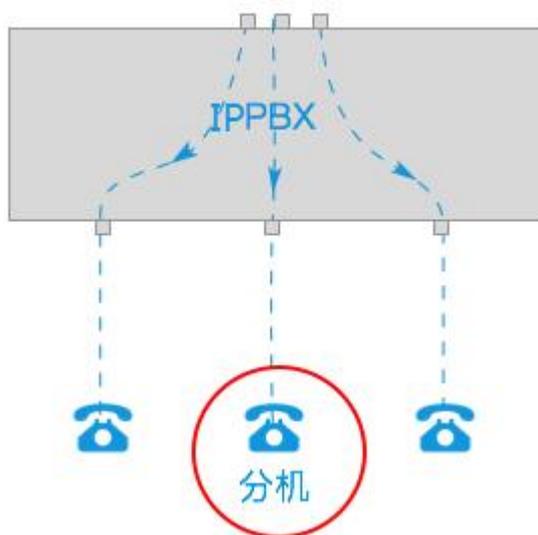


4.2 外线和分机

4.2.1 分机

连接或注册到 IPPBX 上的话机称为分机，分机可以是桌面话机、IP 电话、甚至只是一个软件。

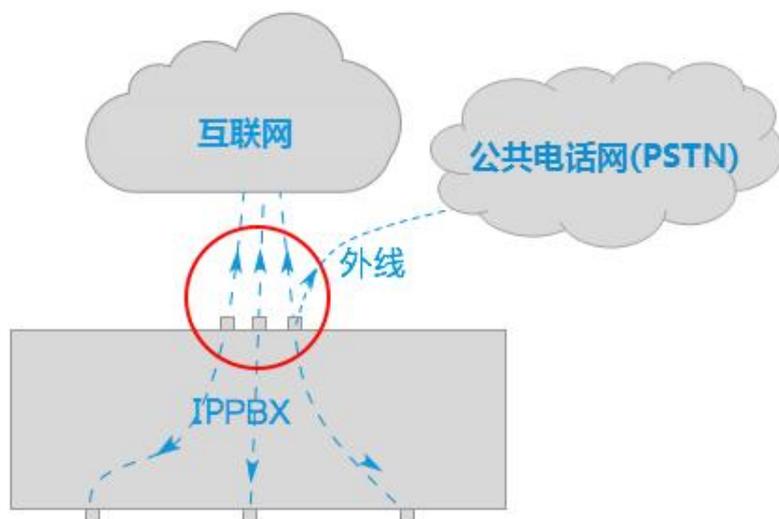
每个分机会有一个短号，称为分机号，分机之间用分机号来呼叫彼此。



4.2.2 外线

IPPBX 和公共电话网相连的线路称为外线，这条线不一定是实际的线路，可能是从电信运营商申请的电话线、光纤、甚至只是一个 IP 账号。

外线也有个号码，称为外线号，外线号一般是从电信运营商申请的。



- 1) **呼入**: 当有人向用固话或手机呼叫 IPPBX 下的某个分机时, 往往要先拨打 IPPBX 的外线号, 等接通后再拨打对应的分机号。当然, IPPBX 也可以将外线和分机一对一绑定, 这样拨打外线号后就不需要拨分机号了, IPPBX 会将呼叫直接送给对应的分机。
- 2) **呼出**: 当分机想要呼叫外面的固话或手机时, 直接拨打对方的号码就可以了, 对方看到的来电号码就是本次外呼用到的外线号码, 而不是你的分机号。

4.3 语法和对象

理解了 OM 及 OM API 的基本概念, 接下来可以了解 OM API 的语法和对象了。

对象:

OM API 主要有 6 个对象, 几乎所有的 API 都围绕着 6 个对象, 因此理解这 6 个对象将会你非常有帮助。



对象描述:

对象名称	概念描述
分机	连接或注册到 OM 上的话机称为分机 (ext)。
外线	OM 和公共电话网相连的线路称为外线或中继 (trunk)。外线可以从电信运营商申请的普通电话线、数字中继线、甚至只是一个 SIP 账号。
来电	从外线呼入的电话称为 来电 (visitor)。

对象名称	概念描述
去电	从外线呼叫的电话称为去电 (outer)。
分机组	分机组 (group) 是多个分机的集合, 当呼叫转接到一个分机组 后, 分机组按照某种策略将呼叫分配给组内的某一个分机。支持的分配策略有 3 种: 顺选、轮选、群震。
语音菜单	语音菜单 (menu) 是一种虚拟通话终端, 通过播放语音提示和检测用户按键来实现一些自助服务的功能。比如: 你拨打 10086 等客服电话时听到的“IVR 语音导航”、“语音验证码”等。

4.4 API 认证

API 认证方式分为 **IP 认证**和**数字签名认证**两种, 您可以根据自己的实际场景选择一种来完成认证。

4.4.1 IP 认证

IP 认证方式, 只允许某一固定 IP 地址向 OM 发送 API 请求, 其他地址统统认为没有权限。

配置参数有两个:

- 1) **服务器地址:** 应用服务器的 IP 地址/域名和监听端口, 如: 192.168.130.27:8989。如果用户未指定端口时默认为 80 端口。
- 2) **URL:** 接收 API 报告的相对路径 (也可不填写)。格式为: {part1}/{part2}/{part3}{.....}, 如: omapi/report。

服务器地址和 URL 组合起来即为应用服务器接收 API 报告的全路径, 如: 192.168.130.27:8989/omapi/report。

应用服务器地址有两个作用:

- 1) **接收 API 报告:** 正如上面介绍的, 用于接收 OM 推送的 API 报告;
- 2) **访问权限控制:** OM 只受理从该服务器的 IP 地址 (端口不影响) 发送的 API 请求; 拒绝受理从其它地址发送的 API 请求, 并对该请求响应 Unauthorized。

配置方法，如下图所示：



4.4.2 数字签名认证

数字签名认证本质上是通过验证 OM 和应用服务器双方持有的密钥来完成认证。

说明：设备接收消息时，优先走 IP 认证，若 IP 认证不通过，再进行数字认证。

[点击这里下载开源代码。](#)

版本要求：

OM 软件版本：Rev 2.1.5.99 及其以上。

适用的应用场景：

- 1) 应用服务器采用动态域名，IP 地址不固定；
- 2) 一个 OM 要对接多个 API 应用服务器；
- 3) API 应用客户端要直接访问 OM；
- 4) API 消息的源地址发生变化。

认证原理：

- 1) 首先，应用服务器和 OM 双方需要配置相同的 API 密码 (API_Password)。
- 2) API_Password 是认证的根本依据。每当 OM 收到一个 API 请求时，都带上一段由 API_Password 生成的认证信息，OM 通过判断请求方的 API_Password 和自己的是否相等，从而判断请求方是否有执行权限。
- 3) 为了保证 API_Password 的网络传输安全，不被黑客通过网络抓包所窃取。所以，我们并不直接在网络上传输 API_Password，而是传输其数字签名。

交互流程:

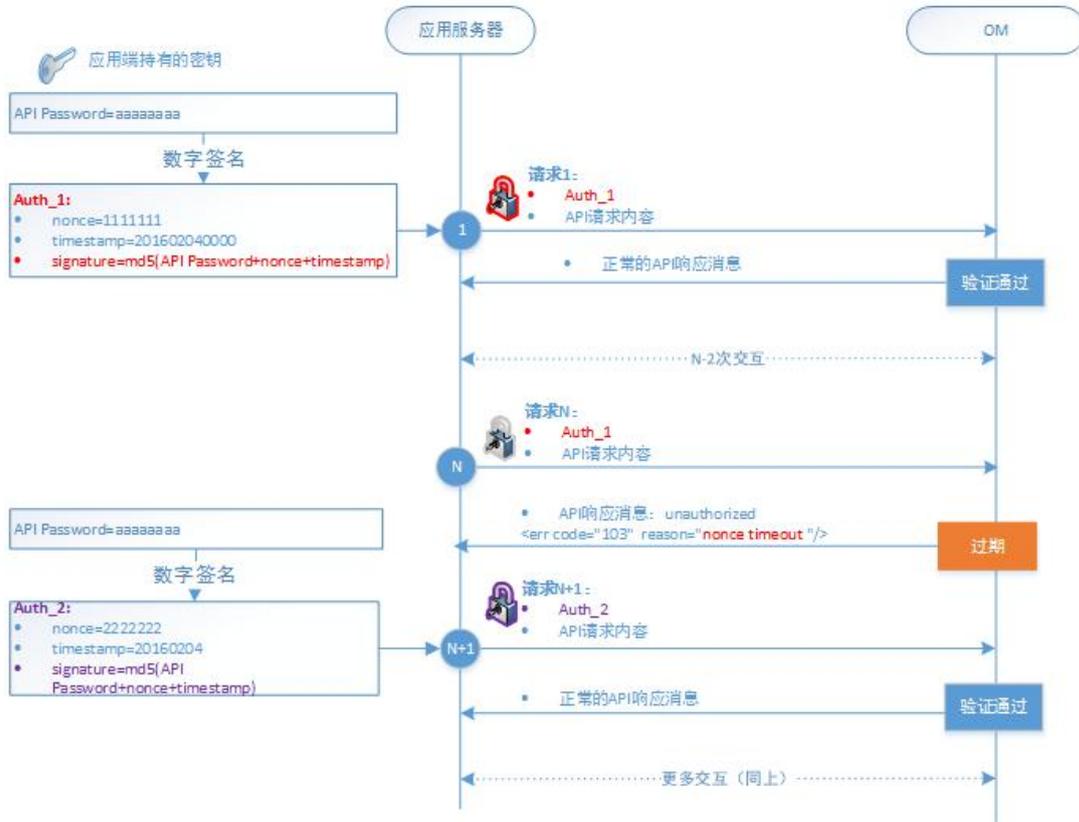
当应用服务器向 OM 发送 API 请求消息, 需要携带上由 API Password 经过数字签名生成的认证信息 (Auth),

Auth 由三部分组成:

- 1) nonce——由应用服务器生成的随机数;
- 2) timestamp——由应用服务器生成的时间戳;
- 3) signature——数字签名, 先将 API Password、nonce、timestamp 三个参数值拼接起来, 然后对其进行 md5 运算得到的 32 位小写字符串。即, $signature = md5(API\ Password + nonce + timestamp)$ 。

- A. 当 OM 收到 API 请求消息后, 用自己的 API_Pasword (此处用 API_Password"表示) 加上【API 请求消息中的 nonce 和 timestamp】进行 md5 运算, 即 $signature" = md5(API_Password" + nonce + timestamp)$, OM 对比 signature" 和 signature 是否相等, 从而推出 API_Password" 和 API_Password 是否相等。如果相等, 则说明请求方有执行权限, 否则没有。
- B. 之后, 为了保证效率, OM 并不要求每次 API 请求都要更新 Auth, 而是会保证在一段有效期 (API_TIMEOUT) 内该 Auth 可以重复使用。
- C. 如果有效期过期或认证信息错误, OM 会给请求方响应 unauthorized 及其错误码。

OM API 数字签名认证的交互流程图



接口说明:

1) 带认证消息的请求消息示例:

```
<?xml version="1.0" encoding="utf-8" ?>
<Auth>
  <Timestamp>1455433892</Timestamp>
  <nonce>14314</nonce>
  <Signature>890b422b75c1c5cb706e4f7921df1d94e69c17f4</Signature>
</Auth>
<Control attribute="Query">
  <DeviceInfo/>
</Control>
```

其中，Auth 结点为认证信息，Control 结点为实际要执行的内容，可根据实际情况更换。

参数说明:

参数名称	类型	参数说明
nonce	string	由应用服务器生成的随机数, 最大长度 32 位。
timestamp	string	应用服务器生成该随机数时的时间戳, 从 1970 年 1 月 1 日 0 点 0 分 0 秒开始到现在的秒数。
signature	string	数字签名, 32 位小写。计算方法为: 将 API Password、nonce、timestamp 三个参数的值按先后顺序拼接成一个字符串并对其进行 md5 运算。其中 API Password 是应用服务器和 OM 双方持有的相同密码。

2) 认证失败的响应消息格式:

```
<?xml version="1.0" encoding="utf-8" ?>
<unauthorized/>
<err code="XXX" reason="XXX"/>
```

参数说明:

Code	Reason	描述说明
100	authentication failed	认证失败。可能原因: API 请求消息中没有携带认证信息。
101	mandatory parameter missing	参数不完整。认证信息必须同时包含 nonce、timestamp、signature 三个参数
102	password validation failure	密码鉴权失败
103	nonce timeout	nonce 过期, nonce 的最大有效期由配置参数 API_TIMEOUT 决定
104	unspecified	其它

配置：

与 OM API 数字签名认证相关的有两个配置参数：API 密码和密码的有效期。

- 1) 这两个参数可在 OM 的 Web 页面上进行配置（web 版本须在 1.0.91_p1 及以上，web 版本查看方法：在地址栏 ip 地址后面输入 version，回车，如 `http://192.168.130.186/version`。
- 2) 若 web 版本较低，可通过在浏览器地址栏输入对应的 URL 接口进行参数查询和配置。

配置界面，如下图所示：



配置参数有四个：

- 1) **服务器地址**：用来接收 OM 推送的消息的服务地址/域名和监听端口，如：
192.168.130.27:8989。如果用户未指定端口时默认为 80 端口。
- 2) **URL**：接收 API 报告的相对路径（也可不填写）。格式为：{part1}/{part2}/{part3}{.....}，
如：omapi/report。
- 3) **API 数字认证密码**：用来认证的密钥，自定义，需和应用服务器持有的密钥保持一致。
- 4) **API 数字认证有效期**：自定义，范围：0~86400，单位：秒，0 表示永久有效。

服务器地址和 URL 组合起来即为接收 API 报告的全路径，如：

192.168.130.27:8989/omapi/report。

4.5 API 反向认证

反向认证，即 OM 向应用服务器推送的 API 消息中携带 Auth 认证信息，应用服务器可根据收到的消息是否满足认证条件来选择是否接收该消息。

版本要求：

Rev 2.1.5.116 及以上。

认证原理：

反向认证方式跟正向数字签名认证方式原理一致，只是密码换成服务器密码。详情可参见 4.3 节数字签名认证原理。

参数：

API 服务器密码：

- 1) 参数名称：API_SERVER_PASSWORD
- 2) 参数 ID：1153
- 3) 参数值：1-16 位的数字、字母。默认值为空

API 数字认证有效期：

- 1) 单位：秒
- 2) 范围：0~86400，默认值：0

注：API 服务器密码暂不支持页面显示，可通过 web API 配置，配置方法参见 6.2 节，值为空时，API 消息中不携带 Auth 信息。

消息示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="BUSY">
<MAC>00:0E:A9:2D:07:D0</MAC>
  <ext id="200" />
</Event>
<Auth>
  <nonce>2E8E52C6</nonce>
  <timestamp>1498542290</timestamp>
  <signature>24fe6019c3f284789f8cfacc3446e7ed</signature>
</Auth>
```

其中，Event 结点为事件信息，Auth 结点为认证信息。

参数说明：

参数名称	类型	参数说明
------	----	------

参数名称	类型	参数说明
nonce	string	随机数，最大长度 32 位。
timestamp	string	时间戳，从 1970 年 1 月 1 日 0 点 0 分 0 秒开始到现在的秒数。
signature	string	数字签名，32 位小写。计算方法为：将 API_SERVER_PASSWORD、nonce、timestamp 三个参数的值按先后顺序拼接成一个字符串并对其进行 md5 运算。其中 API_SERVER_PASSWORD 是应用服务器和 OM 双方持有的相同密钥。

4.6 API 功能开关

（友情提示：超链接返回的快捷键为 Alt + ←）

状态监控：

- 1) 开启状态下，当该线路的状态发生变化或产生通话记录时，OM 会实时向应用服务器推送相应的 API 报告。
- 2) 关闭状态下，当该线路的状态发生变化或产生通话记录时，OM 不会推送相应的 API 报告。

来电应答前控制：

- 1) 开启状态下，当来电呼叫相应的外线（中继）时，OM 不会立刻受理该来电，而是将描述该现象的事件（INVITE 事件）推送给应用服务器，由应用服务器决定是否要受理该来电。如果应用服务器希望 OM 受理该来电，则执行 Accept API 请求；如果希望 OM 拒绝该来电，则执行 Clear API 请求。如果因网络或系统故障等原因应用服务器未收到 INVITE 事件或收到后未执行控制，则 OM 等待超时（等待时间由参数 API_RESPONSE_TO 控制，默认为 5 秒，设置范围 1~120 秒，建议不超过 30 秒）后会自动受理该来电。
- 2) 关闭状态下，当有来电呼叫该外线时，OM 自动受理该来电，不会向应用服务器推送 INVITE 事件。

来电应答后控制:

- 1) 开启状态下, 当有来电呼叫外线时, OM 直接受理该来电, 将其接通并保持静音, 同时将描述该现象的事件 (INCOMING 事件) 推送给应用服务器, 由应用服务器决定如何对该来电进行转接。应用服务器可以执行 Transfer API 请求, 将来电转接到语音菜单、分机或分机组等。如果因网络或系统故障等原因应用服务器未收到 INCOMING 事件或收到后未执行控制, 则 OM 等待超时 (等待时间由参数 API_RESPONSE_TO 控制, 默认为 5 秒, 设置范围 1~120 秒, 建议不超过 30 秒) 后自动将来电转给总机。

特别提示: 仅 Rev2.1.5.112 以后的版本为此机制, 对于之前的版本, OM 等待应用服务器处理超时后仍不会受理该来电, 来电方会一直处于静音状态, 直到电话挂断)。

- 2) 关闭状态下, OM 会安排自身配置对来电进行转接或导航, 如播放欢迎词, 且不会推送 INCOMING 事件。

注: 来电应答前控制和来电应答后控制开关是只针对“来电(即呼入)”而设置的开关, 内部分机互拨及外呼等其他情况则不起作用。

新老 WEB 中 API 开关的对应关系:

老的 web 页面:

模拟分机					
IP 分机		模拟外线		IP 外线	
端口位置	外线号码	API			
1	200	<input type="radio"/> 关闭	<input type="radio"/> 控制	<input checked="" type="radio"/> 监控	
2	201	<input type="radio"/> 关闭	<input type="radio"/> 控制	<input checked="" type="radio"/> 监控	
3	202	<input type="radio"/> 关闭	<input type="radio"/> 控制	<input checked="" type="radio"/> 监控	
4	203	<input type="radio"/> 关闭	<input type="radio"/> 控制	<input checked="" type="radio"/> 监控	
5	204	<input type="radio"/> 关闭	<input type="radio"/> 控制	<input checked="" type="radio"/> 监控	
6	205	<input type="radio"/> 关闭	<input type="radio"/> 控制	<input checked="" type="radio"/> 监控	

新的 web 页面 (OM20/OM50) :



OM20/EOM50 的 web 页面上对 API 开关做了调整, 与老的 web 页面的对应关系为:

线路类型	老的 web 版本	新的 web 版本
分机	API	状态监控
外线 (中继)	监控	状态监控
	控制	状态监控+来电应答后控制 (即两个开关都打开)
	关闭	三个开关都关闭

注: “来电应答前控制”是新增加的控制模式, 老的 web 上没有相应配置项。

API 值说明:

- 1) 状态监控: 开启, API=0x07; 关闭, API=0x00;
- 2) 来电应答前控制: 开启, API=0x10; 关闭, API=0x00;
- 3) 来电应答后控制: 开启, API=0x20; 关闭, API=0x00。

三者可以叠加, 如当某条外线 (中继) 的 API=0x27 时, 表示同时开启了来电应答后控制和状态监控开关。

说明: 2.1.5.94 版本做了修改, 分机配置 API 开关, 0x08 原来是状态监控模式, 94 版本后修改为 DTMF 事件汇报模式, 会汇报单一拨号号码, 呼叫由内部处理。

注: 是否推送 CDR 由参数 API_CDR(id=1058)决定, 而不受 API 开关控制。

API 值对应的事件:



4.7 API 控制模式功能介绍

API 控制模式包括来电应答前控制和来电应答后控制两种模式，两者是前后顺序，可以同时使用。

简单来讲，

- 1) 来电应答前控制决定允许哪些号码呼入；
- 2) 来电应答后控制决定来电接通后执行什么样的流程。

类比图：



可实现的功能：

两种控制模式分别可实现哪些功能，列举几个常见场景：

来电应答前控制模式：

- 1) 来电黑名单：禁止某些骚扰号码呼入；
- 2) 来电白名单(号码绑定固定坐席)：当 VIP 号码呼入时可直接转给特定坐席，而不用听一大堆语音提示。非 VIP 号码按照默认流程执行。

来电应答后控制模式：

- 1) 自定义多级 IVR：如果 OM 自带的 IVR 不能满足您的需求时，就可以自己开发 IVR；
- 2) 自定义来电分配策略：当 OM 默认的顺选 / 轮选 / 群振这三种基本呼叫分配方式不能满足你的实际应用时，可自行开发新策略；
- 3) 自定义来电等待排队策略：如，VIP 优先处理，实时通知前方还有多少人在排队等。

配置：

控制开关的配置是每线的，即每条外线都可以独立配置。

一般情况下，控制开关和状态监控要配合使用。只要用到 API，状态监控开关总是要打开的，控制开关根据实际情况相应打开。



API 功能开关和 API 事件的关系：

- 1) 来电应答前控制，对应 INVITE 事件，即，只有当外线的该开关开启状态下，来电呼叫该外线时 OM 才会向应用服务器推送 INVITE 事件；
- 2) 来电应答后控制，对应 INCOMING 事件，解释同上；
- 3) 状态监控，对应其它所有的事件。

API 功能开关的说明：

API 功能开关相关知识请参考 4.4 节内容。

4.8 外转外功能开关

(友情提示：超链接返回的快捷键为 Alt+ ←)

OM 软件版本 2.1.5.92 以上版本：

外转外权限由系统参数 `API_OUTBOUND_TRANSFER` 确定，当值为 1 时开启，为 0 时关闭。

查询和配置方法：

在浏览器地址栏输入查询或配置的 URL。

参数查询的 URL: `http://OM 的 IP 地址/xml?method=gw.config.get&id=1070&pass=管理员密码`

参数配置的 URL: `http://OM 的 IP 地址/xml?method=gw.config.set&id1070=1&pass=管理员密码`

OM 软件版本 2.1.5.92 以下版本:

配置方法:

步骤 1: 打开浏览器, 在地址栏输入如下格式的路径。

`http://OM 的 web 服务地址
/xml?method=gw.config.set&line_id=1&id700=newrock&id703=123456&id708=O&pass=管理员密码`

注: 该配置的副作用: 第一个分机会成为总机。

步骤 2: 重启设备。

4.9 录音

(友情提示: 超链接返回的快捷键为 `Alt + ←`)

通话录音在企业通信中有很重要的作用。本章主要介绍“OM 如何对通话进行录音”和“如何才能听取录音”。

录音配置和使用说明:

录音方式:

OM 按照将录音文件保存在不同的地方, 把录音方式分为两类:

- 1) **远程录音:** 将录音文件保存到某台电脑上。这台电脑上需要运行一个录音代理软件——全录, 该软件负责实时接收 OM 推送的录音数据包并合成文件;
- 2) **本地录音:** 将录音文件直接保存在 OM 本地。本地录音有两种:
 - A. **外部 USB 设备:** 在 OM 上接个 U 盘等外置 USB 存储设备, 将录音文件保存到上面。
 - B. **内置存储:** 将录音文件保存 OM 的内置存储中。

根据实际情况选择相应的录音方式, 如配置为内置存储录音:



开启分机的录音:

每个分机都有单独的录音开关, 开启后, 对应分机的每次通话都会被录音。

首先, 以管理员身份登录 OM 的管理页面, 分机设置 > 模拟分机/IP 分机, 点击对应分机的最后面的“设置”按钮, 进入分机设置界面, 往下拉, 勾选“录音”, 点击“保存”。如下图所示:



如何判断通话是否被录音:

打开分机的录音开关后, 用分机打一通电话。通话结束后, 观察 CDR 消息中是否有 Recording 字段, 如果有就说明本次通话被录音, 并且生成了录音文件。

CDR 消息如:

```
<?xml version="1.0" encoding="utf-8" ?>
<Cdr id="5520821018163556-0">
```

```

<callid>85</callid>
<visitor id="83" />
<TimeStart>20171116163549</TimeStart>
<Type>IN</Type>
<Route>IP</Route>
<CPN>18201008888</CPN>
<CDPN>200</CDPN>
<TimeEnd>20171116163556</TimeEnd>
<Duration>6</Duration>
<TrunkNumber>02161208234</TrunkNumber>

<Recording>20171116/18201008888_200_20171116_163550_85_cd.wav</Recording
>
</Cdr>

```

查看和获取录音文件：

录音文件可以通过 HTTP 方式来远程查看和获取。

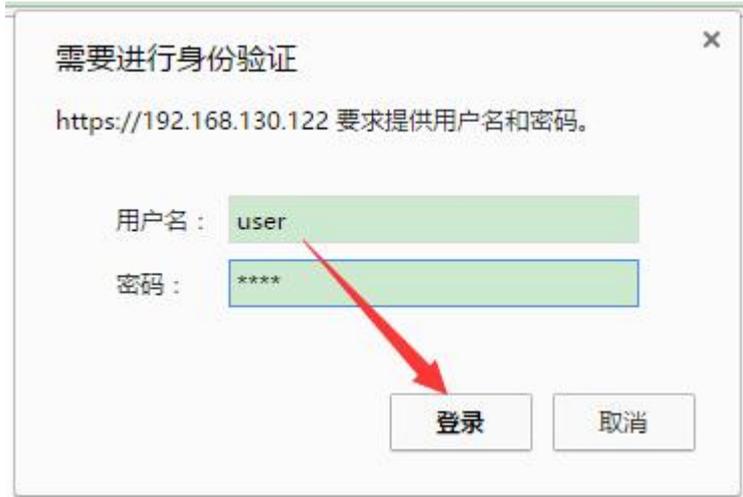
1) 本地录音

本地录音查看方法：

步骤 1： 打开 应用服务 > 存储管理 > 访问地址，查看录音文件的访问地址。



步骤 2： 打开链接，输入用户名密码（默认用户名：user，密码：user），点击登录。



步骤 3: 进入 builtin/Recorder/日期目录，找到对应的日期目录，就可以找到刚才通话生成的录音文件。

本地录音获取方法：

特别提示：

- 1) 本地录音合成需要一定时间，建议在通话结束 10s 后获取录音；
- 2) OM 设备支持的编码方式分为 G729 和 G711(PCMA、PCMU)，G711 编码会生成 pcm 格式录音，设备自动合成 wav 文件存放，可以直接下载播放；而 G729 编码生成的录音设备不会合成，只保留 send 和 recv 两个方向的 dat 文件，不能直接播放，需要下载后本地合成，合成方法请点击[这里](#)查看。

方法 1: HTTP GET 方法下载

获取路径格式为：{OM 的 http 服务地址}/mcc/Recorder/{录音文件的相对路径}

如：

http://192.168.130.45/mcc/Recorder/20150827/211_210_20150827_135634_000a_cg.wav。

注：录音文件相对路径可参考话单中的 Recording 值。



方法 2：通过全录获取本地录音

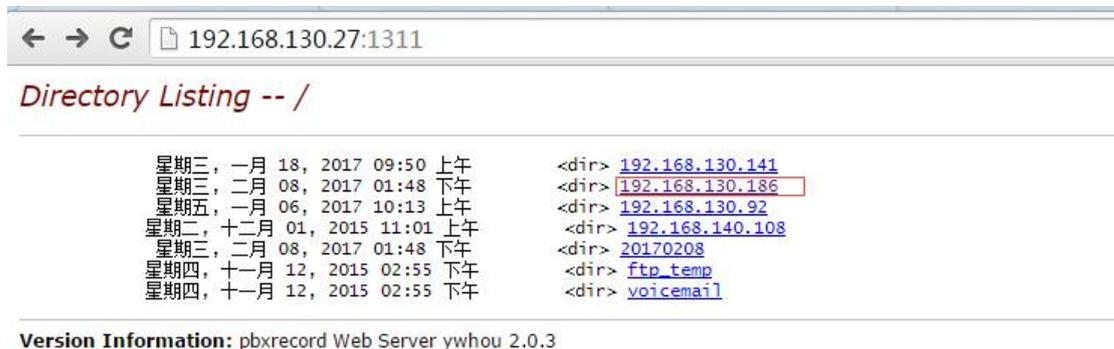
全录录音代理软件支持去本地获取录音文件的功能，版本要求：2.0.5 及以上。

2) 远程录音

录音服务器默认的 HTTP 监听端口为 1311。

步骤 1： 打开浏览器，输入录音服务器的地址，如 `http://录音服务器的 IP 地址或域名:1311`。

步骤 2： 找到对应文件名，文件夹名称是录音服务器的 IP 地址或域名。



注：

- 1) 远程录音文件为全录合成后的文件（默认为 mp3 格式，可设置成 wav），可直接播放。
- 2) 设备 id=698 的参数值须小于 32，否则容易出现全录获取的录音路径和 CDR 中 Recording 字段不匹配的情况。

4.10 语音文件配置说明

(友情提示: 超链接返回的快捷键为 Alt + ←)

OM 的很多对象都需要用到语音文件, 如语音菜单、分机、分机组等。

Voicefile 值的最大长度为: 256 个字符

语音文件的获取方法主要有两种:

- 1) 通过 web 上传;
- 2) 通过 API 远程下载。

4.10.1 Web 上传

格式要求:

Wav: 必须为 8.000 千赫兹, 16 比特或 22.050 千赫兹, 16 比特的单声道.wav 文件, 且文件大小不得超过 1.5MB。

实际上 wav 文件上传完成后, 被 OM 转换成 dat 和 pcm 文件保存到 OM 本地, 即, OM 实际播放的并非 wav 格式的文件, 而是 dat 和 pcm 格式的文件。

制作并上传语音文件:

- 1) 生成语音文件: 点击下载并安装风声软件, 输入要合成的语音文件内容, 合成语音文件(注: 也可以在设备上选择**文字转欢迎词**合成语音, 不过语音质量没有风声合成的好一些);
- 2) web 上传: 登录设备, 点击**基本设置 > 语音管理 > 上传欢迎词**, 上传刚刚合成的语音文件(注: 语音文件名须以 user 开头)。

4.10.2 远程下载

格式要求:

- 1) Dat: G.729 编码格式的 dat 文件;
- 2) Pcm: G.711 编码格式的 pcm 文件。

即, 通过远程下载时, OM 并不对语音文件进行格式转换, 因此不能使用 wav 文件(我们已开发 TTS 服务器, 可实现语音合成以及 wav 到 dat 和 pcm 的格式转换, [点击这里](#)了解详情)。

远程下载方法:

在配置 voicefile 时直接指定下载路径即可，目前支持 http 下载。如：

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
  <menu id="1">
    <voicefile>http://192.168.250.43/test.pcm</voicefile>
  </menu>
</Control>
```

下载规则：

- 1) 如果下载路径中语音文件未加后缀名，则同时下载 dat 和 pcm 文件。
- 2) 如果本地已存在该文件，则 OM 不会重复下载。
- 3) OM 每次启动后，都会检查需要远程下载的哪些语音文件，并自动下载。
- 4) 下载后，录音文件在本地保存路径为 OM 的/tmp/fring 目录下。对于 OM20/OM50 等有内置硬盘的设备，该目录挂载在内置硬盘上，OM 重启文件不会丢失；其对于他无内置硬盘的 OM 设备，该目录挂载在内存中，重启后 OM 会重新下载。

4.11 语音菜单

语音菜单（menu）是一种虚拟通话终端，通过播放语音提示和检测用户按键来实现一些自助服务的功能。比如：你拨打 10086 等客服电话时听到的“IVR 语音导航”、语音验证码等。

4.11.1 基本原理

当一个呼叫和 menu 接通后，

- 1) **播放语音文件：** menu 向通话方播放提前录制好的提示音，比如：“欢迎致电 XXX，查询余额请按 1，业务办理请按 2，停复机管理请按 3，人工服务请按 0”；
- 2) **检测和汇报按键：** 通话方听到提示音后，输入按键信息，menu 检测到按键信息将其汇报给应用服务器。
- 3) **做出响应：** 应用服务器收到按键信息后，根据具体业务情况判断处理，并向 OM 发送对应的 API，比如：将通话方跳转到下一个 menu 或转到分机进行人工服务。

4.11.2 三大特点

- 1) **播放语音：** 可以向通话方自动播放一段提前录制好的语音内容，而不用人为参与。

- 2) **检查和汇报按键**: 可以将通话方输入的按键内容 (DTMF 事件消息) 汇报给开发者开发的应用服务器, 应用服务器根据按键内容对该呼叫进行 菜单之间的跳转、转人工、信息录入等。
- 3) **可同时复用**: menu 是虚拟的, 所以一个菜单可以同时处理很多个呼叫, 各个呼叫相互之间不受任何影响。比如来电 A 转接到 菜单 1 后, 菜单 1 向 A 播放语音并检查 A 按键, 此时如果来电 B 也转接到菜单 1, 菜单 1 又向 B 从头开始播放语音并检查 B 的按键;

4.11.3 应用场景

- 1) **语音外呼**: menu 对外发起呼叫, 实现自动语音外呼、批量外呼等;
- 2) **内部转接**: 对呼叫进行内部转接, 从一个菜单跳转到其它菜单、分机组、分机等, 实现多级 IVR (来电语音导航)、客服质量评分等;
- 3) **呼分机**: 菜单呼叫分机, 实现自动语音叫醒、语音通知等;
- 4) **来/去电转语音信箱**: 客户听语音, 按键, 实现语音导航。

4.11.4 配置

使用 menu 之前需要先进行配置。menu 只能通过 API 进行配置, OM 的管理页面上并没有相应的配置项。

语音菜单的身份标识符 (id) 的有效范围为 1 到 50。

配置的 API:

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
  <menu id="1">
    <voicefile>welcome</voicefile>
    <repeat>3</repeat>
    <infolength>5</infolength>
    <exit>#</exit>
  </menu>
</Control>
```

如需同时配置多个语音文件, 则每个语音文件之间用加号隔开即可, 如: welcome.pcm+connect.pcm, 语音文件的播放顺序为从左到右。如:

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
```

```
<menu id="1">
  <voicefile> welcome.pcm+connect.pcm </voicefile>
</menu>
</Control>
```

也可以用数字加语音文件，如：

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
  <menu id="1">
    <voicefile> 200+welcome.pcm </voicefile>
  </menu>
</Control>
```

参数说明：

一个菜单的配置由三大部分构成：1) 身份标识符、2) 语音文件及其播放规则、3) 按键汇报规则。

1) 身份标识符 (**id**)：使用菜单之前，您需要先进行配置，一台 OM 最多可配置 50 个菜单，一个 id 标示一个菜单，id 有效范围为 1~50，您必须在该有效范围内配置。

2) 语音文件及其播放规则：

- A. 文件名称 (**voicefile**)：想要向通话方默认播放的语音文件名称（该文件需要提前上传到 OM）。
- B. 循环播放次数 (**repeat**)：想要该文件循环播放的次数。

3) 按键汇报规则

- A. 单次汇报按键的最大长度 (**infolength**)：通话方在该菜单下输入的按键达到指定个数后，OM 一次性将这些按键信息汇报给应用服务器，而不必通话方按键一次 OM 就汇报一次。
- B. 输入结束符 (**exit**)：通话方在该菜单下输入了表示已输入完毕的某个指定符号（如：#）后，OM 立刻将之前检查到的按键一次性汇报给应用服务器，而不必等到通话方输入长度达到 **infolength**。

4.11.5 语音文件的播放说明

1) 何时开始播放：

A. 呼叫接通时：菜单发起的呼叫，如：执行完“菜单外呼”和“菜单呼分机”的 API 请求后，需要等待被叫方接通后，OM 才开始自动播放该菜单的语音文件。

B. 呼叫转接到菜单时：已接通的呼叫，如：来电已经呼入到 OM 并且已经接通，此时执行“来电转菜单”的 API 请求，OM 就立刻开始自动该菜单的播放语音文件。

2) 何时停止播放：

A. 完成配置的播放次数：当语音文件的播放次数满足该菜单配置的语音文件播放次数时，停止播放，并向应用服务器汇报语音文件播放完毕的事件消息（EndOfAnn）。

B. 按键打断：当语音播放过程中，通话方输入了按键时，停止播放。

C. 通话中断：当通话方挂断，或呼叫被转接到其他终端，停止播放。

4.11.6 语音菜单相关接口

示例：配置、删除、查询、来电转 menu、去电转 menu、menu 外呼、menu 呼分机。

配置 menu：

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
  <menu id="1">
    <voicefile>welcome</voicefile>
    <repeat>3</repeat>
    <infolength>5</infolength>
    <exit>#</exit>
  </menu>
</Control>
```

删除 menu：

将 menu 配置请清空，就表示删除该 menu。

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
  <menu id="1">
    <voicefile></voicefile>
  </menu>
</Control>
```

查询 menu：

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<Control attribute="Query">
  <menu id="1"/>
</Control>
```

来电转 menu:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <visitor id="41" />
  <menu id="1"/>
  <voicefile>welcome</voicefile>
</Transfer>
```

去电转 menu:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <outer id="14" />
  <outer to="13012345678"/>
</Transfer>
```

menu 呼分机:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <menu id="1"/>
  <ext id="200"/>
</Transfer>
```

menu 呼外部电话:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
  <menu id="1"/>
  <outer to="13012345678"/>
</Transfer>
```

4.12 出队列判断方法

OM 只提供进入队列的事件——QUEUE 事件（包括当前排队数量），出队列需 API 应用服务器自己维护。那么如何判断出队呢？

可根据 RING 事件和 BYE 事件来判断。

- 1) 坐席分机挂断后再次振铃，自动将队列中的通话接入，可根据分机的 **RING** 事件判断出队。

分机 201 振铃，分机 213 被接入：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="RING">
  <ext id="201" />
  <ext id="213" />
</Event>
```

分机 200 振铃，来电 18201008888 被接入：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="RING">
  <ext id="200" />
  <visitor id="75" from="18201008888" to="203" callid="45132" />
</Event>
```

- 2) 分机和来电在排队时主动退出队列，可根据分机和来电的 **BYE** 事件判断出队。

分机 201 主动退出队列：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="BYE">
  <ext id="201" />
  <ext id="3" />
<recording>20170706/201_3_20170706_155709_A00B_cg.wav</recording>
</Event>
```

来电 18201008888 主动退出队列：

```
<?xml version="1.0" encoding="utf-8" ?>
<Event attribute="BYE">
  <visitor id="17" from="18201008888" to="888" callid="40977" />
  <ext id="3" />
</Event>
```

说明：分机和外线在排队时主动退出队列，BYE 事件中 ext id="组号"，这是为了避免和 API 分机组中的 group 字段混淆，所以用 ext id 表示组号。

4.13 插播 TTS 远程语音文件

(友情提示: 超链接返回的快捷键为 Alt + ←)

场景描述:

当对通话方插播的语音不固定(如客户姓名、昵称等)时,可以实时合成远程语音文件,然后下载到 OM 设备本地进行插播。

版本要求:

版本: Rev 2.1.5.117 及以上

实现步骤:

步骤 1: 利用 TTS 接口合成语音文件。

迅时提供合成语音文件的 TTS 接口(仅供测试),可采用其他方式合成。

步骤 2: 将远程 TTS 语音文件下载到 OM 本地。

向 OM 发送配置 menu 的接口即可下载:

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
  <menu id="1">
    <voicefile>http://tts.newrocktech.com:9880/wav/20170203/1486091400698.p
cm</voicefile>
    <repeat>1</repeat>
    <infolength>3</infolength>
    <exit>#</exit>
  </menu>
</Control>
```

注: <voicefile>中为通过 TTS 合成的远程语音文件路径。

这样,便把语音文件 1486091400698.pcm 下载到了 OM 本地。

步骤 3: 执行语音插播命令。

如,对来电插播语音:

```
<?xml version="1.0" encoding="utf-8" ?>
<Transfer attribute="Connect">
```

```
<visitor id="15" />
<voicefile>1486091400698</voicefile>
</Transfer>
```

注：<voicefile>中为下载后的语音文件名称，文件名必须以字母开头。

更多语音插播接口，可[点击这里](#)查看。

4.14 跨域访问 OM

4.14.1 简介

HTML5 新的标准中，增加了“Cross-Origin Resource Sharing”特性，这个特性的出现使得跨域通信只需通过配置 http 协议头即可解决。

关于跨域的推荐资料：<http://www.alloyteam.com/2012/11/html5-cors/>。

4.14.2 限制条件

设备：OM20、OM50（注：其它 OM 型号的版本尚未发布）。

软件版本要求：Rev 2.1.5.100 及以上。

注：如果低于该版本，请不要尝试以下操作，会导致异常。如需升级版本，请联系迅时技术支持电话：4007779719 或 4006172700。

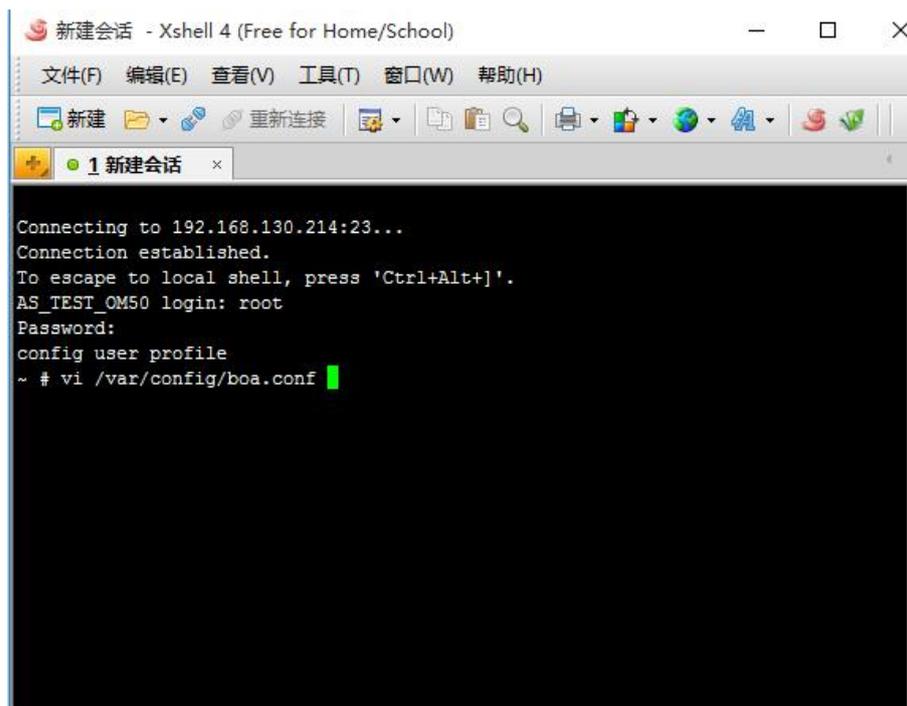
4.14.3 解决方法

OM 设备支持该跨域方式，但默认并没有开启，需要用户 Telnet 登录 OM 后台修改配置文件。

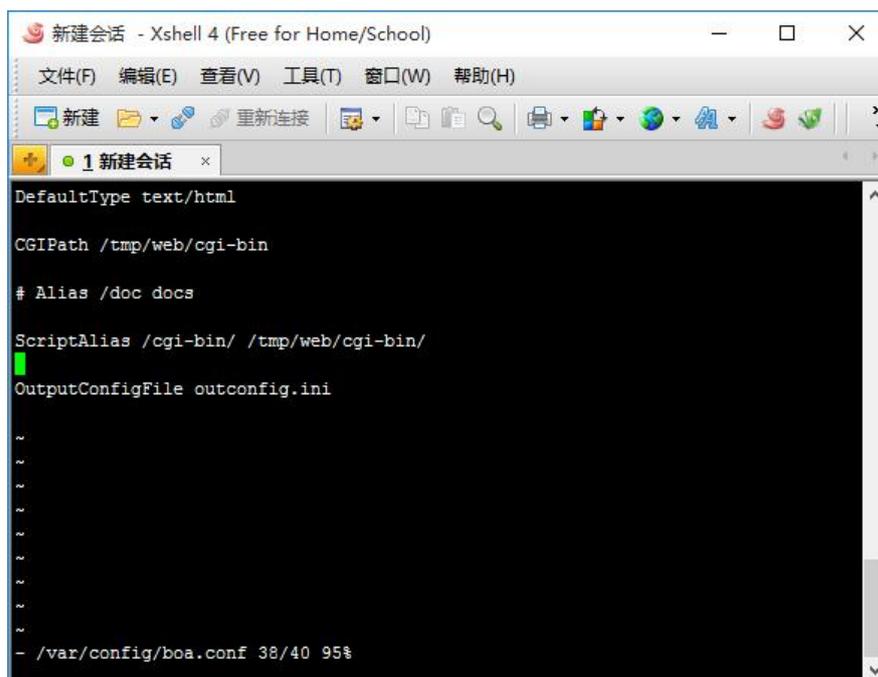
操作步骤如下：

- 1) telnet 登录到设备后台（telnet 方法介绍）。
- 2) 修改配置。

步骤 1：执行命令 vi /var/config/boa.conf。

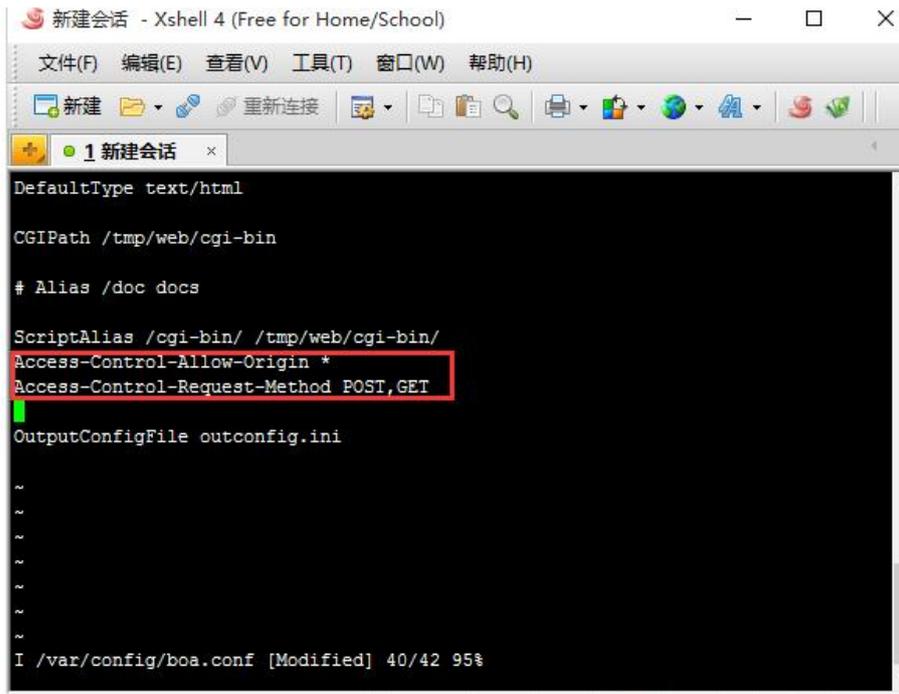


步骤 2: 通过方向键移动光标到文件尾“OutputConfigFile outconfig.ini”的前一行。



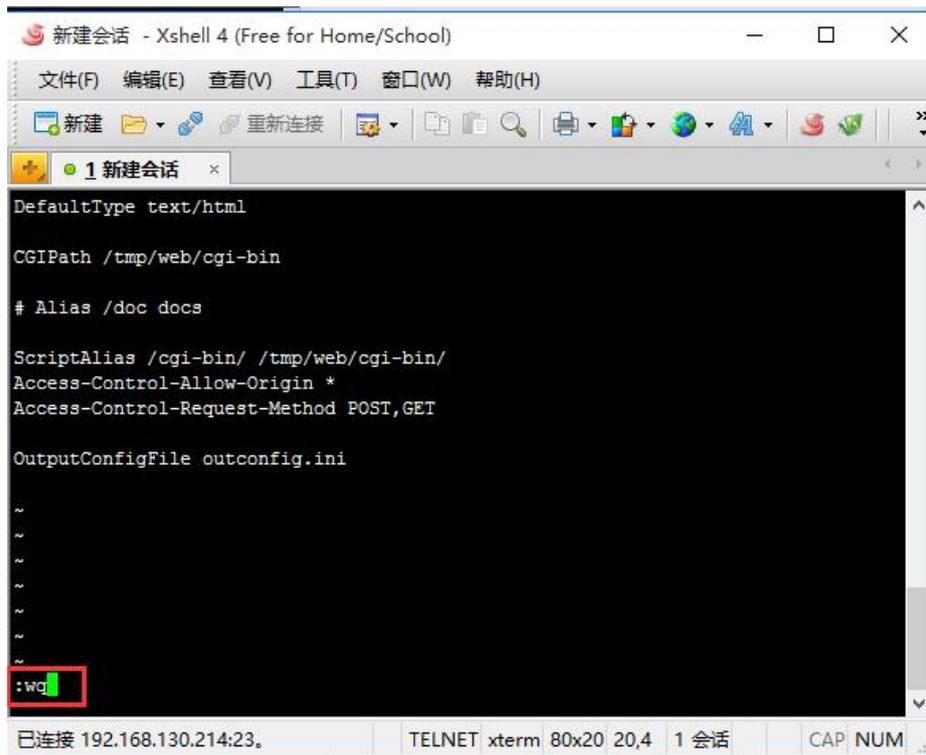
步骤 3: 按 i, 进入插入模式, 并插入以下两个配置项。

```
Access-Control-Allow-Origin      *
Access-Control-Request-Method    POST,GET
```



```
新建会话 - Xshell 4 (Free for Home/School)
文件(F) 编辑(E) 查看(V) 工具(T) 窗口(W) 帮助(H)
新建 重新连接
1 新建会话
DefaultType text/html
CGIPath /tmp/web/cgi-bin
# Alias /doc docs
ScriptAlias /cgi-bin/ /tmp/web/cgi-bin/
Access-Control-Allow-Origin *
Access-Control-Request-Method POST,GET
OutputConfigFile outconfig.ini
~
~
~
~
~
I /var/config/boa.conf [Modified] 40/42 95%
```

步骤 4: 按 `esc` (退出键), 退出插入模式, 然后按 `:wq`, 最后按回车键退出并保存文件。



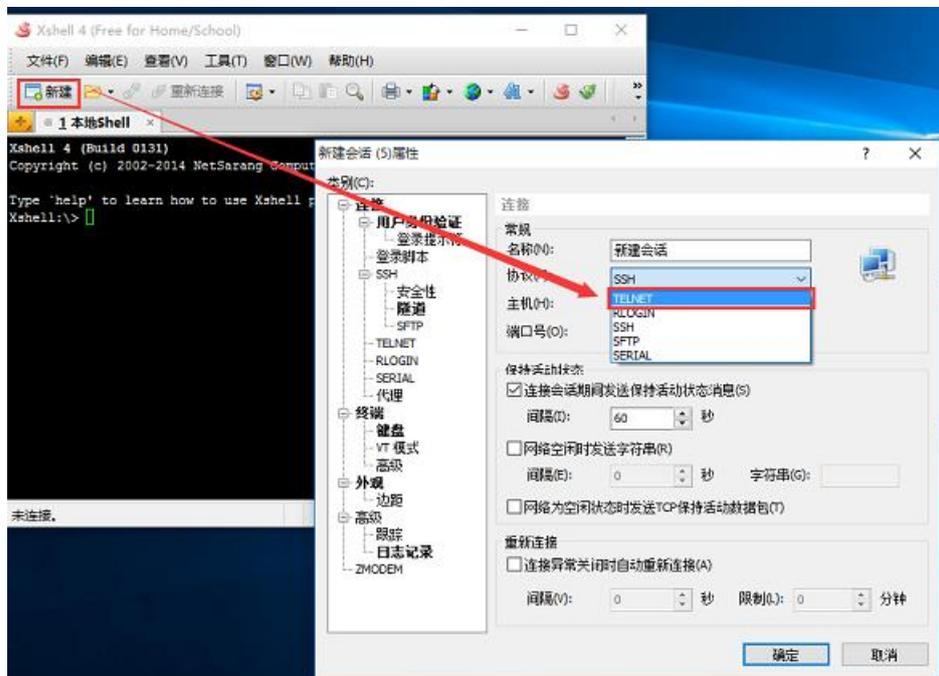
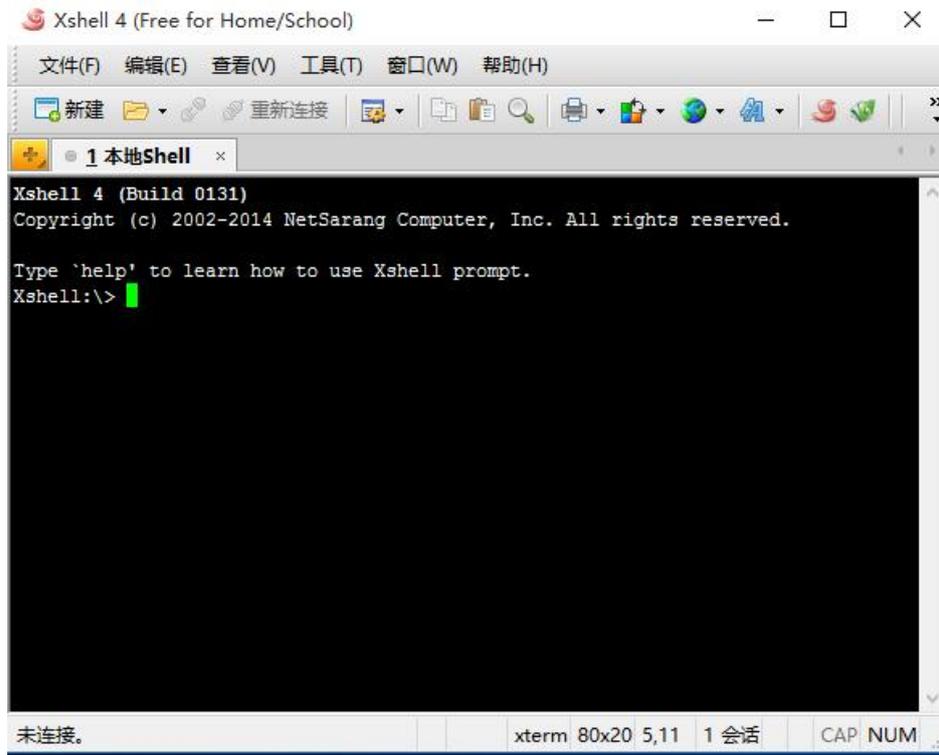
```
新建会话 - Xshell 4 (Free for Home/School)
文件(F) 编辑(E) 查看(V) 工具(T) 窗口(W) 帮助(H)
新建 重新连接
1 新建会话
DefaultType text/html
CGIPath /tmp/web/cgi-bin
# Alias /doc docs
ScriptAlias /cgi-bin/ /tmp/web/cgi-bin/
Access-Control-Allow-Origin *
Access-Control-Request-Method POST,GET
OutputConfigFile outconfig.ini
~
~
~
~
~
:wq
已连接 192.168.130.214:23. TELNET xterm 80x20 20,4 1 会话 CAP NUM
```

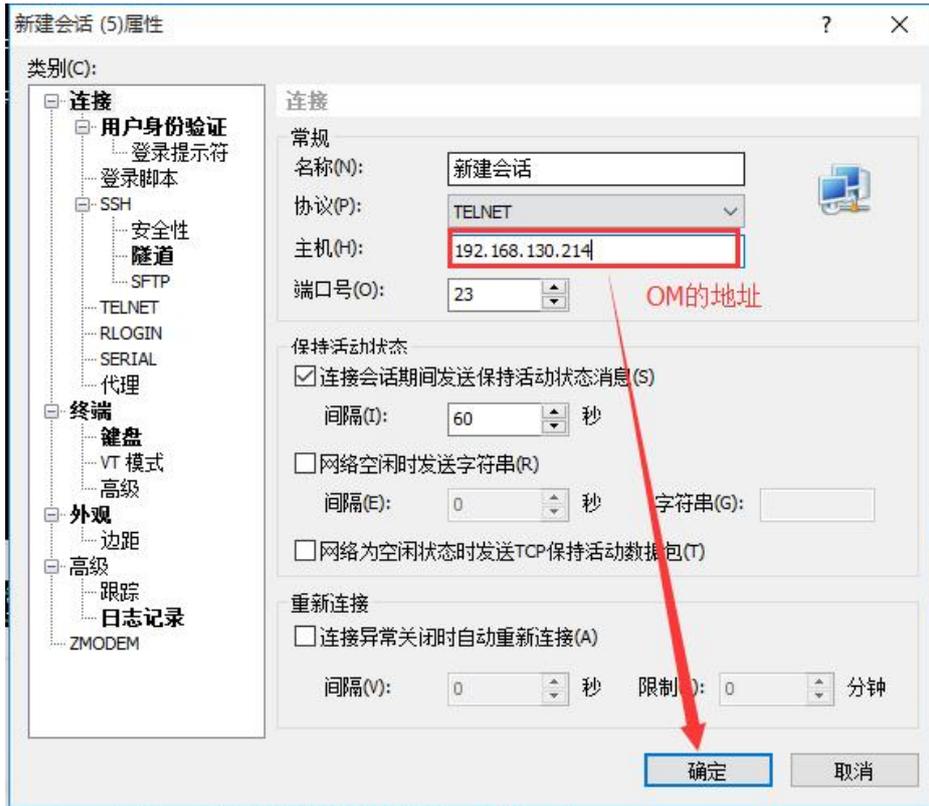
步骤 5: 重启 `boa` 组件。

步骤 6: 执行命令 `killall -9 boa`, 重启 `boa` 组件。

步骤 3: 新建 Telnet 会话。

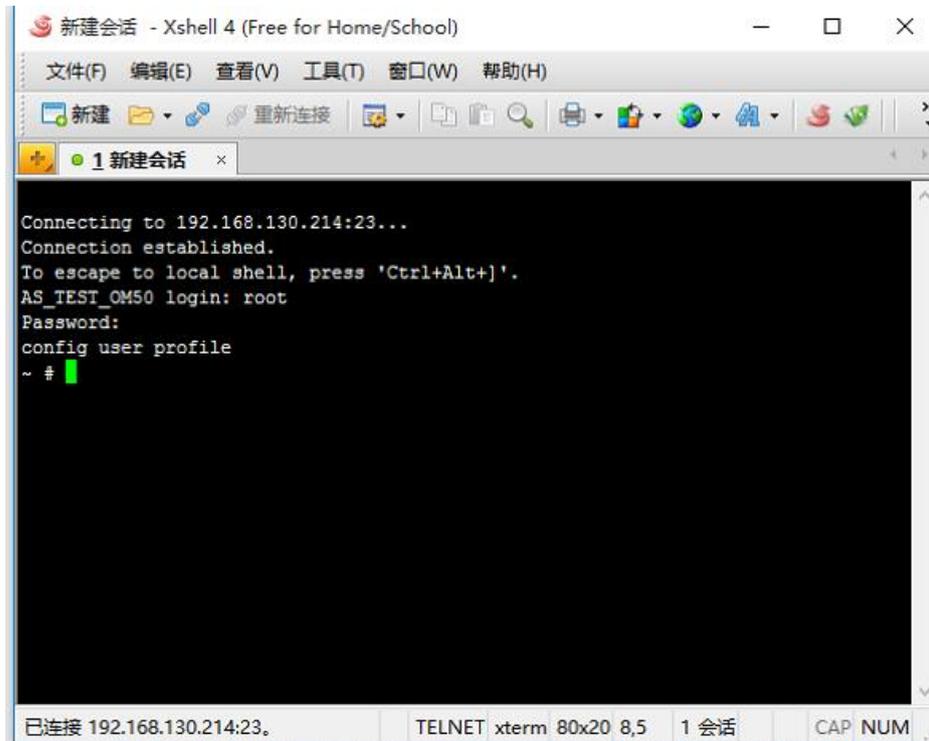
打开 Xshell，点击“新增”，弹出新的窗口，其中“协议”选择 telnet 协议，“主机”填写 OM 的 IP 地址。点击“确定”，然后点击“连接”。





步骤 4: Telnet 登录。

输入用户名和密码。其中, 用户名为 root, 密码为刚才在设备 web 页面上设置的密码。(△: 输入密码不显示)



步骤 5：后台操作。

通过以上步骤即完成 Telnet 登录 OM 设备后台，OM 后台为 Linux 系统，可执行常用的 Linux 命令。

步骤 6：查看日志。

OM API 日志可以到 /var/log/debug.log 查看，日志级别需打到“6（RTP 事件）”。

SIP 信令日志可到 /var/tmp/message.log 查看。

5 常见问题解答

本章介绍了 OM API 常见问题及解决方法。

5.1 网络环境相关问题解答

- 1) **OM 设备重启后，API 服务器可以正常收到 OM 发送的 API 消息，但过一段时间后却收不到 API 消息了，再后来又收到 API 消息了，这是什么原因导致？**

问题原因：API 监听服务器和 OM 发送消息时采用的 TCP 连接方式不一致。

问题分析：

- A. 若 OM 采用短连接方式推送，则每发送一条消息都会新建一个 TCP 连接，若 API 监听服务器采用长连接，即收到消息后未执行断开连接，如此 OM 不停的新建 TCP 连接，最终会导致 TCP 资源不足。
- B. 若 OM 采用长连接方式推送，当发送数据包包含多条消息时，若 API 监听服务器采用短连接，即收到一条消息后就断开此连接，那么其他的消息也就丢失了。

注：只有 OM 推送消息时支持长连接方式，接收消息时只支持短连接，故应用服务器向 OM 发送消息时必须采用短连接方式。

5.2 API 认证相关问题解答

- 1) **API 认证失败，返回 Unauthorized，怎么处理？**

解答：

返回 Unauthorized，说明应用服务器地址未通过 OM 的 API 认证。您可以参考 API 认证介绍，检查您的配置是否正确。

- 2) **应用服务器地址是否支持域名？**

解答：

支持域名。

- A. 对于 IP 认证，应用服务器域名应为静态域名，用来完成认证，同时用来接收 OM 推送的 API 消息。

- B. 对于数字签名认证，控制服务器（即发送消息的服务器）的域名可以为动态域名，而监听服务器（即用来接收 API 消息的服务器）的域名须为静态域名。

注：不论是 IP 认证还是数字认证，在设备上配置的应用服务器地址应为某一固定 IP 或静态域名。

5.3 menu 相关问题解答

- 1) 为什么配置的语音文件不能播放，或者有时候播放正常，有时候又不正常？

问题原因：

通话采用的语音编码方式（G729、G711 等）会选择播放不同格式（dat、pcm）的语音文件，所以，如果配置的 voicefile 指定了文件格式，而通话采用的编码方式与之不匹配时，就会导致不能播放的情况。

解决方法：

- A. 配置 voicefile 时，不要加文件后缀；
- B. 要保证 OM 本地两种格式的 voicefile 都存在；

注：通过 web 上传的 wav/dat 格式的语音文件，系统会自动转换成两种格式，通过 API 远程下载的不会自动转换。

5.4 配置相关问题解答

- 1) 如何配置分机的免打扰功能？配置完是否需要重启？

解答：

- A. 配置分机的接口中携带<No_Disturb>yes</No_Disturb>即可开启免打扰功能，详情请参考配置分机的接口。
- B. 通过 API 执行查询、配置和转接等命令后都不需要重启设备。
- C. 设备重启后，分机的免打扰功能失效（主要是为了避免设备重启后呼叫异常问题），需重新配置。

- 2) 配置了分机的免打扰，但返回消息中无免打扰字段，且查询分机时也无免打扰字段，怎么查看是否配置成功？

解答:

若配置的分机为总机, 会出现这种情况, 只有普通分机可配置免打扰功能, 总机不支持免打扰。

3) 配置分机组时, 只添加了两个分机, 为什么返回结果中多了很多分机?

解答:

是因为该分机组中已有其他分机, 再配置时只会累加而不会删除其他分机。

- A. 若想删除分机组内部分分机, 需利用配置分机的接口, 将其中的<group>节点去掉。
- B. 若想删除分机组内所有分机, 可采用以下接口:

```
<?xml version="1.0" encoding="utf-8" ?>
<Control attribute="Assign">
  <group id="XX"/>
</group>
</Control>
```

5.5 呼叫相关问题解答

1) 为什么点击拨号分机振铃后会自动挂断?

问题原因:

呼叫方式若为先呼主叫后呼被叫, 点击拨号时, 主叫摘机后会选择一种话机本身和 OM 都支持的编码方式, 若被叫不支持这种编码方式, 则呼叫无法建立。

解决方法:

可以通过屏蔽 OM 的响应编码方式或调整编码顺序来解决。

注: 点击拨号方式分为先呼主叫后呼被叫、先呼被叫后呼主叫两种方式, 可通过修改参数 API_CALLING 来配置。

2) 双向外呼失败, 怎么办?

解答:

- A. 首先确保外线资源够用, 即外线个数为 2 个及以上;
- B. 其次确保外转外开关为开启状态, [点击这里](#)查看外转外权限开启方法。

注：来电转外部电话、去电转外部电话、语音菜单外呼、双向外呼（回拨）时，都需要开启**外转外**权限开关。

3) 可以指定中继外呼吗？

解答：

可以指定中继外呼。

- A. 加前缀外呼，外呼时随机选择任意一条空闲的模拟/IP 外线，此时要求设备拨号规则也设置为加前缀外呼；
- B. 在外呼命令中增加<trunk id>节点，可指定从某条固定中继呼出，版本要求：Rev2.1.5.109 及以上；
- C. 注： [点击这里查看更多详情](#)。

4) 如何获取 visitor id？

解答：

开启状态监控和来电应答前/来电应答后控制开关后，当有来电时，OM 会向应用服务器推送 INVITE/INCOMING 事件（这两个事件中都会有 visitor id），从而获取 visitor id。

注：具体开启来电应答前控制开关还是来电应答后控制开关可根据实际情况而定，也可同时打开。

5.6 事件相关问题解答

1) 为什么收到的呼叫状态事件(如, RING/ALERT 等)没有 visitor/outer id 和 callid 等信息？

问题原因：

因为外线的 API 没有开启，导致分机不能获取外线信息。

解决方法：

开启外线的 API 功能开关。

注：监控和控制模式都可以，根据具体情况选择。

2) 外呼的时候，没有接通就挂断了，为什么会收到 ANSWERED 事件，而且 CDR 的通话时长大于 0？

问题原因:

- A. 因为模拟外线没有开启反极性信号导致。
- B. 通过模拟外线外呼时，模拟外线并不能检测到对方是否接通，且默认为一旦呼叫被送出就立马接通了。所以，会收到 ANSERED 事件，且不管对方接通不接通，本次通话结束后，CDR 的通话时长都将大于 0。

注：此问题仅仅在通过模拟外线外呼时才会出现。

解决方法:

如果你使用的模拟外线已通过电信运营商开通了**反极性信号**的业务，则可以通过 OM 的 web 页面开启对应线路的**反极性检测**开关，即可判断对方是否接通。

重要说明：如果未通过电信运营商开启**反极性信号**，请勿开启设备的**反极性检测**开关，否则，会造成即使对方摘机了也没有 ANSERED 事件，CDR 的通话时长将一直为 0。

3) Event 中的 callid 为什么和 CDR 中的不一致？callid 为什么不唯一？我如何判断通话的唯一性？

问题原因:

- A. 对于用户而言，A 和 B 的通话虽然表示是一路，但对于 OM 而言，实际上是由很多段通话组合起来的，目的是方便 OM 对通话的任意节点进行拆分和转接（如将来电和分机 A 的通话转给分机 B 等）。
- B. 具体细节，你可以不去考虑，知道不要用 callid 作为通话的唯一标识写入数据库即可。

解决方法:

- A. visitor id 和 outer id 从通话开始到结束的时间内是唯一的，但会被回收循环利用，可作为临时唯一标识。
- B. 在第一次收到 visitor/outer id 时，将当前的时间戳与 visitor/outer id 建立一对一的映射关系，用时间戳保证唯一性（常见应用有：通话过程中的便签记录，需要和其通话记录一一绑定）。
- C. 当通话结束后，cdr id 是唯一的，可以写入数据库作为通话的唯一标识。另外，cdr 中同样会包含 visitor/outer id，再次通过和时间戳的映射关系，找到 cdr 和时间戳的映射关系。（对应通话便签记录的应用方式：最终实现通话便签记录内容和通话记录的绑定）。

4) 什么时候判断通话被接通?

解答:

- A. 对于来电, 当收到分机的 Answer 事件时, 说明通话接通。
- B. 对于去电, 当收到 OM 推送的 Answered 事件时, 说明通话接通。

说明: 若有 Alert 事件说明成功呼叫, 对方回铃, 若无 Alert 事件, 说明对方号码为空号或其他原因导致呼叫失败。

5.7 CDR 相关问题解答

1) 一通电话, 产生好几个类型的话单, 应该以哪个通话记录或录音为准?

解答:

有些情况下, 一通电话可能存在多个 CDR 的情况。尤其是模拟中继呼入。而这些 CDR 中, 可能有些是用户不需要的, 可以过滤掉不保存。

分机的 CDR (用户需要关心)

- A. 呼入: TYPE 为 IN, CPN 为外部电话, CDPN 为分机, TrunkNumber 为中继号码。
- B. 呼出: TYPE 为 OU, CPN 为分机, CDPN 为外部电话, TrunkNumber 为中继号码。
- C. 分机呼分机 (主叫): TYPE 为 LO, CPN 为分机, CDPN 为分机, TrunkNumber 为空。
- D. 分机呼分机 (被叫): TYPE 为 IN, CPN 为分机, CDPN 为分机, TrunkNumber 为空。
- E. 呼叫转移入: TYPE 为 FI, CPN 为主叫, CDPN 为分机 (这个 CDR 说明有一通电话本来是呼叫分机的, 结果被呼叫转移走了, 原主叫和分机的转移方之间通话了多久)。

模拟中继和 IP 中继的 CDR (用户可忽略)

- A. 呼入: TYPE 为 LO, CPN 是外部电话, CDPN 为分机。
- B. 呼入: TYPE 为 IN, CDPN 等于 TrunkNumber 为中继号码。
- C. 呼出: TYPE 为 OU, CPN 等于 TrunkNumber 为中继号码, CDPN 为外部电话。
- D. 呼叫转移出: TYPE 为 FW, CPN 为分机, CDPN 为转移后的号码。 (这个 CDR 说明有一通电话被呼叫转移出局, 并且通话了多久)。

常见场景下话单情况

- A. 模拟中继呼入转分机：生成 2 个 CDR，一个 type 为 IN，一个 type 为 LO，IN 的为分机的 CDR，LO 的为外线的 CDR（注：用户可忽略掉 LO 的 CDR）。
- B. IP 中继呼入转分机：生成 2 个 CDR，type 都为 IN，一个是分机的 CDR，一个是外线的 CDR。
- C. 分机通过模拟中继呼出：1 个 CDR，TYPE 为 OU（注：如果有两个 OU 的 CDR，需要升级软件到 86 及其以上版本）。
- D. 分机通过 IP 中继呼出：1 个 CDR，TYPE 为 OU。

2) 为什么有的话单（CDR）中没有录音文件信息？录音和 CDR 怎么对应上？

解答：

- A. CDR 中<Recording>节点记录录音文件路径信息，如果没有开启中继或分机的录音开关，其话单中就没有 Recording 字段，也就没有录音文件路径。
- B. 将录音文件名称与<Recording>节点信息相比较来判断录音和 CDR 是否相对应。

5.8 其他问题

1) API 对设备和话机有没有要求？

解答：

- A. 所有 OM 设备的 API 都是一样的，只是版本不同，MX 设备不支持 API。
- B. API 对话机没有要求。

2) 设备存储数据吗？录音文件怎么落地的？

解答：

- A. OM 触发 API 消息时，会自动把消息推送给应用服务器，由应用服务器存储，如通话记录、事件。
- B. 录音包括远程录音和本地录音，本地录音可以将录音文件存储在设备内置存储或 U 盘，远程录音可配合迅时开发的全录软件来保存、管理录音。

6 附录

本章介绍 OM 设备系统参数、功能开关、release 版本。

6.1 系统参数列表

(友情提示: 超链接返回的快捷键为 Alt + ←)

参数名称	参数 id	参数值	说明
API_MAC	349	yes, no 默认值 no	yes: OM 发送 API 消息时会带 OM 的 MAC 地址, 格式为: xxxxxx; no: 不会携带 MAC 参数。
API_RESPON SE_TO	526	1-120 秒 默认值 5 秒	OM 向应用服务器推送 API 报告(如: INVITE 事件)后, 如果应用服务器在 API_RESPONSE_TO 时间内未响应, 则 OM 将自动处理。
CONTROL_TY PE	554	1, 0 默认值 0	1: OM 推送 API 报告采用 TCP 长连接 0: OM 推送 API 报告采用 TCP 短连接 注: 长连接时, OM 可能会建立多条连接, 并保持部分连接, CDR 和呼叫消息采用不同的连接, 若网络异常或应用服务器断开了连接, OM 会重新建立连接; 短连接时, OM 每次发送消息都会新建连接, 应用服务器收到消息后应断开连接。

参数名称	参数 id	参数值	说明
FT_API_1	599	0x00, 0x07, 0x17, 0x27, 0x37 默认值为 0x00	API 开关参数。 0x00: 关闭 0x07: 状态监控 0x17: 来电应答前+状态监控 0x27: 来电应答后+状态监控 0x37: 来电应答前+来电应答后+状态监控 注: 该参数为 每线参数 , 点击这里查看 每线参数查询和配置方法。
CB_TYPE	615	0, 1 默认值 0	0: 双向外呼第一路呼叫从 IP 中继呼出; 1: 双向外呼第一路呼叫从模拟中继呼出。
API_DISPLAY	624	yes, no 默认值 yes	yes: 发起的呼叫会用 display 中定义的号码作为主叫; no: 发起的呼叫用 IP 中继的号码。 注意: 该功能需运营商支持。
CONFIG_CHANGE	883	0, 2 默认值 0	0: 当通过 web 修改 OM 语音相关的参数值时, 不触发 CONFIG_CHANGE 事件; 2: 当通过 web 修改 OM 语音相关的参数值时, OM 会向应用服务器推送 CONFIG_CHANGE 事件。
API_CALLING	978	yes, no 默认值 no	yes: 分机点击拨号先呼叫主叫, 主叫摘机后再呼被叫; no: 先呼被叫, 被叫回铃再呼主叫。

参数名称	参数 id	参数值	说明
FT_ALERT_IN FO_X	986	on, off 默认 off	自动应答参数 (X 为线路号)。 on: 开启自动应答; off: 关闭自动应答。 注: 开启后 INVITE 呼叫 IP 分机时携带 Alert-Info 字段, 若终端支持则自动摘机应答。
API_CPN	991	yes, no 默认值 no	yes: API 控制 CPN, 替换主叫号码; no: API 显示 API_DISPLAY。 注意: 该功能需运营商支持。
API_CDR	1058	yes, no 默认值 yes	yes: 推送 CDR; no: 不再推送 CDR。
API_OUTBOU ND_TRANSFE R	1070	0, 1 默认值 0	0: 不允许通过 API 将两个外部电话进行转接; 1: 允许通过 API 将两个外部电话进行转接。
API_TIMEOUT	1088	0-86400 秒 默认值 0	数字签名认证超时时间参数, 0: 表示永久生效; 1-86400: 表示超时后需重新认证。
API_PASSWO RD	1089	1-16 位 的数字/ 字母 默认值为 空	数字签名认证密码。 注: API 认证优先选择 IP 认证, IP 认证不通过时走数字签名认证。
API_METHOD	1125	0, 1 默认 0	API HTTP 消息推送方式参数。 0: 使用 HTTP GET 方式; 1: 使用 HTTP POST 方式。

参数名称	参数 id	参数值	说明
API_SERVER_PASSWORD	1153	1-16 位的数字/字母 默认值为空	数字签名认证密码（发送）。 注：该密码必须和应用服务器接收消息时要求的密码一致。
API_FLAG	1198	0x00, 0x01, 0x04 默认值 0x00	0x00: API 的配置不会写入系统的配置文件，OM 重启后配置丢失； 0x01: API 呼叫，被叫只汇报 ring 事件，不再汇报 busy； 0x04: API 的配置会写入系统的配置文件。 注：用来防止 API 命令过多写配置文件，进行双重保护。

6.2 系统参数查询和配置方法

（友情提示：超链接返回的快捷键为 Alt+ ←）

OM API 有一些系统参数并没有开放到 web 页面和 API 接口，需要通过 URL 修改。

6.2.1 查询方法

打开浏览器，在地址栏输入 URL 路径：

- 1) 针对一般参数，URL 格式为：

http://设备 IP/xml?method=gw.config.get&id=XX&pass=admin(设备管理员密码)。

- 2) 针对每线参数，URL 格式为：

http://设备 IP/xml?method=gw.config.get&line_id=X&id=XX&pass=admin(设备管理员密码)。

注：

- 1) line_id 为线路编号。
- 2) 系统参数分为一般参数和每线参数，具体参数信息可参见 6.1 节参数列表。

示例图分别如下所示：



注：如果提示，item value="not defined"，则原因可能为：1) 参数名有误 2) OM 的软件版本太低。

6.2.2 配置方法

在浏览器地址栏输入如下格式的 URL 进行配置：

- 1) 针对一般参数，URL 格式为：

http://设备 IP/xml?method=gw.config.get&idXX=XXX&pass=admin(设备管理员密码)。

- 2) 针对每线参数，URL 格式为：

http://设备 IP/xml?method=gw.config.get&line_id=X&idXX=XXX&pass=admin(设备管理员密码)。

示例图分别如下所示：





6.2.3 响应结果

- 1) 如果 `rsp stat="restart"`，配置成功，重启设备后生效。
- 2) 如果 `rsp stat="ok"`，配置成功。
- 3) 如果 `rsp stat="fail"`，配置失败。