菊风业务框架

VoIP Client

发表日期: 2015-07-02

访问 http://www.juphoon.com 了解更多产品信息

菊风 VoIP SDK 用户开发指南

菊风系统软件有限公司 http://www.juphoon.com 电话: +86-574-87287820 传真: +86-574-87304379

Copyright © 2015, Juphoon System Software Corporation.

版权所有

Juphcon

日米

1.	产品	概述	9
1	.1	用途	9
1	.2	阅读对象	9
1	.3	内容简介	9
1	.4	缩略词和缩写词	9
1	.5	参考	9
2.	软件	设计	.10
	0 1	安白禮梅加	10
2	 	合广 地 但未	10
2	<u>~</u> > 3		.10
2	 2 4	小名参教型黑	
•			
3.	LICE	:NSE	.11
3	3.1	LICENSE激活和回收	.11
	3.1.1	· 接口和回调	.11
	3.1.2	License 激活流程	. 12
	3.1.3	License 解绑流程	.13
	3.1.4	4 License 激活和解绑示例代码	.13
4.	客户	端管理	.15
2	l.1	初始化和销毁	.15
	4.1.1	·····································	.15
	4.1.2	2 初始化操作过程	.15
	4.1.3	3	.16
2	1.2	启动和停止	.16
	4.2.1	· <i>接口与回调</i>	.16
	4.2.2	? 启动操作过程	.16
	4.2.3	3 <i>停止操作过程…</i>	.17
2	1.3	参数设置	.17
	4.3.1	<i>音频设置</i>	.18
2	1.4	样例代码	.19
	4.4.1	9 <i>单帐号初始化</i>	.19
5.	用户	管理	.19
5	5.1	账号管理	.19
-	5.1.1	———— — <i>用户接口</i>	.20
	5.1.2	? <i>读取账号列表</i>	.20
	5.1.3	3 <i>获取当前账号</i>	.21
	5.1.4	4 创建新账号	.21
	5.1.5	5 <i>删除账号</i>	.22
5	5.2	打开与关闭	.22

52.2 打开操作过程 53 登話与注销 53.3 登話与注销 53.2 发起登话边理 53.3 双问贯站边理 53.4 发起注销过程 53.5 发起注销过程 53.6 发起注销过程 53.7 发起注销过程 53.8 发动注销过程 53.6 发动注销过程 53.7 发动管理 54.1 发口与回调 52.2 潜音呼叫 53.6 紫动注销过程 53.7 发起呼叫过程 53.8 双问呼叫过程 53.9 探师呼叫过程 53.1 发起呼叫过程 531 发起呼叫过程 531 发起呼叫过程 531 发起呼叫过程 531 发起呼叫过程 532 发出呼呼叫过程 533 取得呼叫过程 535 终止呼叫道理 536 终上呼叫道理 537 修士/> 538 终上/> 541 理過時 541 把始呼叫道理 535 修士/ 641 平可以前年 642 運士/ </th <th>5.2.1</th> <th>用户接口</th> <th>22</th>	5.2.1	用户接口	22
52.3 关闭操作过程 5.3 登陆与注销 5.3.1 接口与回调 5.3.2 发起置私过程 5.3.3 取得置私过程 5.3.4 发起注销过程 5.3.5 被动注销过程 5.3.6 被动注销过程 5.3.6 被动注销过程 6.1 接口与回调 6.2 适音即叫 6.2 适音即叫 6.2 适音即叫 6.2 【意即叫 6.2 【意即叫 6.2 【意印叫 6.3 【初時呼叫过程 6.4 并以得到显示 6.3 【初時呼叫过程 6.3 【初時呼叫过程 6.3 【想要呼叫过程 6.3 【認時呼叫过程 6.3 【認時呼叫过程 6.3 【想要呼叫过程 6.4 評叫前時 6.4 評叫前時 6.4 評叫前時 6.4 評叫前時 6.4 不可以前時 6.5 呼叫说時 6.6 呼叫说時 6.7 呼叫限制	5.2.2	打开操作过程	23
5.3 登站与注销	5.2.3	关闭操作过程	23
5.3.1 度口与回调 5.3.3 取消费拉过程 5.3.4 发起注험过程 5.3.5 放动注韵过程 5.3.5 放动注韵过程 6.1 按口与回调 6.2 语音呼叫 6.2 语音呼叫 6.2 接听呼叫过程 6.2 接听呼叫过程 6.2.3 现消呼叫过程 6.4 挖地呼叫过程 6.2.4 挖地呼叫过程 6.2.5 炎止呼叫过程 6.2.6 炎止呼叫过程 6.2.7 发送DTMF过程 6.3 视频呼叫 6.3 初期呼叫 6.3 发起呼叫过程 6.3 提起呼叫过程 6.3 集的中叫过程 6.3 第四時中 6.4 平叫前转 6.4 不可及前转 6.4 不可及前转 6.5 呼叫零時 6.6 呼叫取制 6.7.1 呼风波中 6.8 呼叫取制 6.7.1 停入超止 6.8 呼叫取制 6.7 可叫取制 6.8 呼叫转移 6.9 铃音描加 6.9 铃音描加 6.9 铃音描加 6.9 铃音描加 6.9 铃音描加 6.9 公	5.3 🔮	登陆与注销	24
5.3.2 发起登航过程 5.3.3 取得最航过程 5.3.4 发起产精过程 5.3.5 旅航注销过程 6. 会话管理 6.1 6.1 接口与回调 6.2 富音呼叫 6.2.1 发起呼响过程 6.2.2 发师呼叫过程 6.2.3 取得呼叫过程 6.2.4 だ夢呼叫过程 6.2.5 终止呼叫过程 6.2.6 主以呼呼叫过程 6.2.7 发送了「加」过程 6.3 视频呼叫 6.3.1 发起呼响过程 6.3.2 提供呼叫过程 6.3.3 取得時回过程 6.3.4 把您時可叫过程 6.3.5 终止呼呵叫过程 6.3.4 您必可可以过程 6.3.4 您必可可以过程 6.3.4 您可可以过程 6.4.1 无张件前转 6.4.2 遵何以有 6.4.4 不可及前转 6.4.5 呼叫成向 6.4.6 考出所前 6.7 呼叫成南 6.7.1 原入组上 6.8 呼叫操制 6.7.1 原入組上 6.8.1 算能 <	5.3.1	接口与回调	24
5.3.3 取消登航过程 5.3.4 发起注销过程 5.3.5 被动注销过程 6.1 弦口与回调 6.1 弦口与回调 6.2 语音呼叫 6.2 语音呼叫 6.2.1 发起呼叫过程 6.2.2 接听呼叫过程 6.2.3 取消呼叫过程 6.2.4 だ地呼叫过程 6.2.5 终止呼叫过程 6.2.6 注却得過显示 6.2.7 发送DTMF过程 6.3 视频呼叫过程 6.3 视频呼叫过程 6.3 观频呼叫过程 6.3 观频呼叫过程 6.3 观频呼叫过程 6.3 观频呼叫过程 6.3 双频呼叫过程 6.3 双频呼叫过程 6.3 投影呼叫过程 6.3 发起呼叫过程 6.3 发音师说过程 6.3 发音师说过程 6.3 发音师说程 6.4 平明前转 6.4 平明前转 6.4 不可及前转 6.5 呼叫零特 6.6 呼叫保持 6.7 呼叫限制 6.7 呼叫限制 6.7 可以限制 6.7 呼叫限制 6.8 回归转转 6.9 铃音描放 6.1 查转 6.2 经防转 6.3 经营转 6.4 章表 6.5 呼叫我 6.6.1 章转 6.8 平时转者 6.9 铃音描放	5.3.2	发起登陆过程	25
5.3.4 发起注册过程 5.3.5 拔动注销过程 6.1 按口与回调 6.2 语音呼叫 6.2 语音呼叫 6.2 接听呼叫过程 6.2.3 取消呼叫过程 6.2.4 拒绝呼叫过程 6.2.5 终止呼叫过程 6.2.6 生却号码显示 6.2.7 发送DTMF过程 6.3 初频呼叫 6.3 初期呼叫过程 6.3.1 发起呼叫过程 6.3.2 接听呼叫过程 6.3.3 取得呼叫过程 6.3.4 拒绝呼叫过程 6.3.5 终止呼叫过程 6.3.6 袋止呼叫过程 6.3.7 发送DTMF过程 6.3.8 取得呼叫过程 6.3.1 发起呼叫过程 6.3.2 提你呼叫过程 6.3.3 取得呼叫过程 6.3.4 拒绝呼叫过程 6.3.5 终止呼叫过程 6.4.1 无穷仲前转 6.4.2 選忙前转 6.4.3 无应答前转 6.4.4 不可以前转 6.5 呼叫该特 6.5 呼叫该特 6.6 呼叫或特 6.7 呼叫取得 6.7.1 呼入阻止 6.8 呼叫转移 6.8.1 盲转 6.8.2 資助转 6.9 铃音播放 6.9 铃音播放 6.9 铃音播放 6.9 铃音描放 6.9 铃音播放 6.9 容音響 6.9 公司转 <td>5.3.3</td> <td>取消登陆过程</td> <td>26</td>	5.3.3	取消登陆过程	26
5.3.5 被动注朝过程 6. 全话管理 6.1 按口与国潮 6.2 语音呼叫 6.2 语音呼叫 6.2.1 发起呼叫过程 6.2.3 取消呼叫过程 6.2.4 拒绝呼叫过程 6.2.5 终止呼叫过程 6.2.6 生叫号母显示 6.2.7 发送OTME过程 6.3 视频呼叫 6.3 视频呼叫 6.3 视频呼叫 6.3 视频呼叫 6.3 視频呼叫 6.3 視频呼叫 6.3 視频呼叫 6.3 現場呼叫过程 6.3.3 取消呼叫过程 6.3.4 拒绝呼叫过程 6.3.5 紫化声呼叫过程 6.3.4 拒绝呼叫过程 6.3.5 紫化声响过程 6.4 呼叫前转 6.4.1 无条件前转 6.4.1 无条件前转 6.4.3 无政密韵转 6.4.4 不可及韵转 6.5 呼叫等特 6.6 呼叫快转 6.7 呼叫限制 6.7 呼叫限制 6.7 「可限関制 6.7 「可以関制 6.8.1 直转 6.8.2 咨询转 6.9 铃音潮放 6.1 直转 6.2 咨询转 6.3 早期繁体 6.4.1 查求時 6.5 呼叫的转 6.6 呼叫转 6.7 呼叫转 6.8 回归	5.3.4	发起注销过程	27
6. 会话管理 6.1 按口与回调 6.2 语音呼叫 6.2.1 发起呼叫过程 6.2.2 按听呼叫过程 6.2.3 取消呼叫过程 6.2.4 拒绝呼叫过程 6.2.5 经止呼叫过程 6.2.6 主叫号跟宏示 6.2.7 发送DTNF过程 6.3 视频呼叫 6.3 视频呼叫 6.3.1 发起呼叫过程 6.3.1 发起呼叫过程 6.3.3 取消呼叫/过程 6.3.4 拒绝呼叫过程 6.3.5 终止呼叫过程 6.3.4 拒绝呼叫过程 6.3.5 终止呼叫过程 6.3.6 终止呼叫过程 6.3.7 死营前转 6.4 呼叫前转 6.4 呼叫前转 6.4.1 无条件前转 6.4.2 選忙前转 6.4.5 呼叫送待 6.6 呼叫送待 6.6 呼叫误待 6.7 呼叫限制 6.7 呼叫限制 6.7.1 原入阻止 6.8.1 盲转 6.8.1 盲转 6.8.2 咨询时转 6.8.1 盲转 6.8.1 盲转 6.8.2 咨询时结 6.9 铃音描放 6.9 铃音播放 6.9 只會書做次 6.9 只會書做次 6.9 只要書做次 6.9 只要書做次 6.9 只要書做次 6.9 只要書做次	5.3.5	被动注销过程	28
6.1 技口与回调	6. 会话管	理	28
6.2 语音呼叫 6.2.1 发起呼叫过程 6.2.2 接听呼叫过程 6.2.3 取消呼叫过程 6.2.4 拒绝呼叫过程 6.2.5 终止呼叫过程 6.2.6 差叫号码显示 6.2.7 发达DTM-过程 6.3 视频呼叫 6.3 视频呼叫过程 6.3 取消呼叫过程 6.3 取消呼叫过程 6.3 取消呼叫过程 6.3 取消呼叫过程 6.3 取消呼叫过程 6.3 取消呼叫过程 6.3.4 拒绝听叫过程 6.3.5 终止呼叫过程 6.4.6 非恒响转 6.4.1 元条件前转 6.4.1 元条件前转 6.4.3 无应答前转 6.4.4 不可及前转 6.4.5 呼叫故向 6.4.6 年注册前转 6.7 呼叫限制 6.7 呼叫限制 6.7 呼叫限制 6.7 呼叫限制 6.8.1 盲转 6.8.1 盲转 6.8.2 咨询转 6.9 铃音瑞放 6.9 铃音瑞放 6.9 铃音瑞放 6.9 铃音瑞放 6.9 铃音瑞放 6.9 经音谐数 6.9 经音谐数 6.9 经音谐数 6.9 全音谐数 6.9 全音谐数 6.9 公司转 6.9 全音谐数 6.9 全音谐数 6.9 章音谐数	6.1 招	安口与回调	29
6.2.1 发起呼叫过程 6.2.2 接听呼叫过程 6.2.3 取消呼叫过程 6.2.4 拒绝呼叫过程 6.2.5 终止呼叫过程 6.2.6 左叫号码显示 6.2.7 发送DTMF过程 6.3 视频呼叫 6.3.1 发起呼叫过程 6.3.2 接听呼叫过程 6.3.3 职消呼叫过程 6.3.4 拒绝呼叫过程 6.3.5 终止呼叫过程 6.3.4 拒绝呼叫过程 6.3.5 终止呼叫过程 6.4.6 牙燃炉时过程 6.4.7 无条件前转 6.4.8 Funin转 6.4.4 不可及前转 6.4.5 炉叫说向 6.4.6 未注册前转 6.6 呼叫操制 6.7.1 呼入阻止 6.8 呼叫操制 6.7.1 呼入阻止 6.8 Full 6.8.1 直转 6.8.2 咨询转 6.9 铃音播放 6.9.1 本地铃音 6.9.2 早期媒体 6.10 会注定时器	6.2 겵	与音呼叫	32
62.2 按听呼叫过程	6.2.1	发起呼叫过程	
62.3 取消呼叫过程 62.4 拒绝呼叫过程 62.5 终止呼叫过程 62.6 主叫号码显示 62.7 发送OTMF过程 6.3 视频呼叫 6.3 视频呼叫过程 6.3 视频呼叫过程 6.3 视频呼叫过程 6.3 视频呼叫过程 6.3 机炉呼叫过程 6.3 非绝呼叫过程 6.3 非绝呼叫过程 6.4 呼叫前转 6.4 呼叫前转 6.4 呼叫前转 6.4 不可及前转 6.4 不可及前转 6.4 不可及前转 6.4 不可及前转 6.4 不可及前转 6.4 标明前转 6.4 不可及前转 6.4 呼叫前转 6.4 呼叫前转 6.4 原则读向 6.4 未注册前转 6.5 呼叫读向 6.6 呼叫保持 6.7 呼叫限制 6.7 呼叫限制 6.7 呼叫限制 6.8 停叫转移 6.8 停叫转移 6.9 铃音播放 6.9 铃音播放 6.9 铃音播放 6.1 查接 6.2 咨询转 6.9 袋音播放 6.9 名声播放 6.1 查接 6.2 名前转 6.2 名前转 6.3 专利的常 6.4 专利的学 6.5 年	6.2.2	接听呼叫过程	
62.4 拒绝呼叫过程 62.5 终止呼叫过程 62.6 主叫号码显示 62.7 发送DTMF过程 6.3 视频呼叫 6.3.1 发起呼叫过程 6.3.2 按听呼叫过程 6.3.3 取消呼叫过程 6.3.4 拒绝呼叫过程 6.3.5 终止呼叫过程 6.3.4 拒绝呼叫过程 6.3.5 终止呼叫过程 6.3.6 终止呼叫过程 6.3.7 近季前转 6.4 呼叫前转 6.4 可以前转 6.4.1 无条件前转 6.4.2 遇忙前转 6.4.3 无应答前转 6.4.4 不可及前转 6.4.5 呼叫收向 6.4.6 未注册前转 6.5 呼叫等待 6.6 呼叫保持 6.7 呼叫限制 6.8.1 盲转 6.8.2 咨询转 6.8.1 盲转 6.8.2 咨询转 6.9 铃音播放 6.9 铃音播放 6.9.1 本地铃音 6.9.2 早期媒体	6.2.3	取消呼叫过程	34
62.5 终止呼叫过程 62.6 主叫号码显示 62.7 发送DTMF过程 6.3 视频呼叫 6.3 视频呼叫 6.3.1 发起呼叫过程 6.3.2 按所呼叫过程 6.3.3 取消呼叫过程 6.3.4 拒绝呼叫过程 6.3.5 经比呼叫过程 6.3.4 拒绝呼叫过程 6.3.5 经比呼叫过程 6.4 呼叫前转 6.4 呼叫前转 6.4.1 无条件前转 6.4.2 遇忙前转 6.4.3 无应答前转 6.4.4 不可及前转 6.4.5 呼叫这向 6.4.6 未注册前转 6.5 呼叫转待 6.6 呼叫保持 6.7 呼叫限制 6.7 呼叫限制 6.8.1 盲转 6.8.2 咨询转 6.9 铃音播放 6.9 铃音播放 6.9.1 本地铃音 6.9.2 早期線体 6.10 会话定时器	6.2.4	拒绝呼叫过程	
6.2.6 主叫号码显示 6.2.7 发送DTMF过程 6.3 视频呼叫 6.3.1 发起呼叫过程 6.3.2 接听呼叫过程 6.3.3 取消呼叫过程 6.3.4 拒绝呼叫过程 6.3.5 终止呼叫过程 6.3.6 停吁叫前转 6.4 呼叫前转 6.4.1 无条件前转 6.4.2 週忙前转 6.4.3 无应答前转 6.4.4 不可及前转 6.4.5 咿叫改向 6.4.6 未注册前转 6.6.6 呼叫误持 6.7 呼叫限制 6.7 呼叫限制 6.7 呼叫限制 6.8.1 盲转 6.8.1 盲转 6.8.2 咨询转 6.9 铃音播放 6.9 铃音播放 6.9.1 本地铃音 6.9.2 早期媒体 6.10 会话定时器	6.2.5	终止呼叫过程	35
6.2.7 发送DTMF过程 6.3 视频呼叫 6.3.1 发起呼叫过程 6.3.2 接听呼叫过程 6.3.3 取消呼叫过程 6.3.4 拒绝呼叫过程 6.3.5 终止呼叫过程 6.3.6 终止呼叫过程 6.3.7 无应答前转 6.4 呼叫前转 6.4.1 无条件前转 6.4.2 遇忙前转 6.4.3 无应答前转 6.4.4 不可及前转 6.4.5 咿叫改向 6.4.6 未注册前转 6.6 呼叫误持 6.7 呼叫限制 6.7 呼叫限制 6.7 呼叫限制 6.8 哼叫转移 6.8 信 6.8 复数 6.8.1 直转 6.8.2 咨询转 6.9 铃音播放 6.9 铃音播放 6.9.1 本地铃音 6.9.2 早期媒体 6.10 会话定时器	6.2.6	<i>主叫号码显示</i>	
6.3 视频呼叫 6.3.1 发起呼叫过程 6.3.2 接听呼叫过程 6.3.3 取消呼叫过程 6.3.4 拒绝呼叫过程 6.3.5 终止呼叫过程 6.3.6 终止呼叫过程 6.4 呼叫前转 6.4.1 无余件前转 6.4.2 邊忙前转 6.4.3 无应答前转 6.4.4 不可及前转 6.4.5 呼叫改向 6.4.6 未注册前转 6.5 呼叫转待 6.6 呼叫味持 6.7 呼叫限制 6.7.1 呼入阻止 6.8 呼叫转移 6.8.1 盲转 6.8.2 咨询转 6.9 铃音播放 6.9.1 本地铃音 6.9.2 早期媒体 6.10 会话定时器	6.2.7	发送DTMF过程	
6.3.1 发起呼叫过程 6.3.2 接听呼叫过程 6.3.3 取消呼叫过程 6.3.4 拒绝呼叫过程 6.3.5 终止呼叫过程 6.4.1 元朱件前转 6.4.1 元朱件前转 6.4.2 遇忙前转 6.4.3 无应答前转 6.4.4 不可及前转 6.4.5 呼叫改向 6.4.6 未注册前转 6.5 呼叫转待 6.6 呼叫保持 6.7 呼叫限制 6.7.1 呼入阻止 6.8 呼叫转移 6.8.1 盲转 6.8.2 咨询转 6.9 铃音播放 6.9 铃音播放 6.9 早前爆体 6.10 会话定时器	6.3 礼	见频呼叫	
6.3.2 接听呼叫过程	6.3.1	发起呼叫过程	
6.3.3 取消呼叫过程 6.3.4 拒绝呼叫过程 6.3.5 终止呼叫过程 6.4 呼叫前转 6.4.1 无条件前转 6.4.2 遇忙前转 6.4.3 无应答前转 6.4.4 不可及前转 6.4.5 呼叫改向 6.4.6 未注册前转 6.5 呼叫等待 6.6 呼叫限制 6.7.1 呼入阻止 6.8 呼叫转移 6.8.1 盲转 6.8.2 咨询转 6.9 铃音播放 6.9.1 本地铃音 6.9.2 早期媒体 6.10 会话定时器	6.3.2	接听呼叫过程	39
6.3.4 拒绝呼叫过程	6.3.3	取消呼叫过程	39
6.3.5 终止呼叫过程	6.3.4	拒绝呼叫过程	40
6.4 呼叫前转 6.4.1 元条件前转 6.4.2 週忙前转 6.4.3 元应答前转 6.4.3 元应答前转 6.4.4 不可及前转 6.4.5 呼叫改向 6.4.6 未注册前转 6.5 呼叫等待 6.6 呼叫保持 6.7 呼叫限制 6.7.1 呼入阻止 6.8 呼叫转移 6.8 呼叫转移 6.8.1 盲转 6.8.2 咨询转 6.9 铃音播放 6.9 铃音播放 6.9 全音播放 6.9 全音播放 6.9 全音振媒体 6.10 会话定时器	6.3.5	终止呼叫过程	41
6.4.1 无条件前转	6.4 R	乎叫前转	42
6.4.2 週忙前转. 6.4.3 无应答前转. 6.4.4 不可及前转. 6.4.4 不可及前转. 6.4.5 呼叫改向. 6.4.6 未注册前转. 6.5 呼叫等待 6.6 呼叫保持. 6.7 呼叫限制. 6.7 呼叫限制. 6.7 呼叫限制. 6.8 呼叫转移. 6.8 呼叫转移. 6.8.1 盲转. 6.8.2 咨询转. 6.9 铃音播放. 6.9.1 本地铃音. 6.9.2 早期媒体. 6.10 会话定时器.	6.4.1	无条件前转	42
6.4.3 无应答前转	6.4.2	遇忙前转	42
6.4.4 不可及前转	6.4.3	无应答前转	42
6.4.5 呼叫改向	6.4.4	不可及前转	42
6.4.6 未注册前转	6.4.5	呼叫改向	42
6.5 呼叫等待 6.6 呼叫保持 6.7 呼叫限制 6.7.1 呼入阻止 6.8 呼叫转移 6.8.1 盲转 6.8.2 咨询转 6.9 铃音播放 6.9.1 本地铃音 6.9.2 早期媒体 6.10 会话定时器	6.4.6	未注册前转	43
 6.6 呼叫保持	6.5 P	乎叫等待	43
 6.7 呼叫限制	6.6 P	乎叫保持	44
 6.7.1 呼入阻止 6.8 呼叫转移	6.7 R	乎叫限制	45
 6.8 呼叫转移	6.7.1	呼入阻止	45
6.8.1 盲转 6.8.2 咨询转 6.9 铃音播放 6.9.1 本地铃音 6.9.2 早期媒体 6.10 会话定时器	6.8 P	乎叫转移	46
6.8.2 咨询转 6.9 铃音播放 6.9.1 本地铃音 6.9.2 早期媒体	6.8.1	盲转	46
 6.9 铃音播放 6.9.1 本地铃音 6.9.2 早期媒体 6.10 会话定时器 	6.8.2	咨询转	47
6.9.1 <i>本地铃音</i>	6.9 铭	令音播放	48
6.9.2 <i>早期媒体</i>	6.9.1	本地铃音	48
6.10 会话定时器	6.9.2	早期媒体	48
	6.10 ≨	< <p>会话定时器</p>	50

6.11 拨号	ት URI	50
6.11.1	SIP URI	50
6.11.2	TEL URI	50
6.11.3	模拟POTS终端业务定制	50
6.12 会词	举例	51
6.12.1	基本呼叫过程	51
6.12.2	取消呼叫过程	52
6.12.3	拒绝呼叫过程	53
6.12.4	呼叫改向过程	53
6.12.5	<i>呼叫保持过程</i>	54
6.12.6	<i>呼叫转移过程</i>	55
7. 媒体录制	存储	57
7.1 音视	频录制存储接口	57
7.1.1 🛃	安口	57

表目录

表 1-1 缩略词和缩写词	9
表 3-1 License 激活和回收接口	11
表 3-2 License 回调处理	12
表 3-3 客户端 License 激活过程说明	13
表 3-4 客户端解绑回收过程说明	13
表 3-5 license 激活和回收示例代码	15
表 4-1 客户端初始化和销毁接口	15
表 4-2 客户端初始化过程说明	15
表 4-3 客户端销毁过程说明	16
表 4-4 客户端启动和停止接口	
表 4-5 客户端启动过程说明	17
表 4-6 客户端停止过程说明	17
表 4-7 音频输入设备接口	
表 4-8 音频输出设备接口	
表 5-1 用户管理接口	20
表 5-2 读取账号列表说明	21
表 5-3 获取当前账号说明	21
表 5-4 创建新账号说明	21
表 5-5 删除账号过程说明	
表 5-6 打开和关闭接口	22
表 5-7 用户打开过程说明	23
表 5-8 用户关闭过程说明	23
表 5-9 登陆与注销用户接口	24
表 5-10 登陆与注销 GUI 回调	24
表 5-11 发起登陆过程说明	25
表 5-12 取消登陆过程说明	26
表 5-13 发起注销过程说明	27
表 5-14 被动注销过程说明	
表 6-1 会话用户接口	
表 6-2 会话 GUI 回调	
表 6-3 发起呼叫过程说明	
表 6-4 接听呼叫过程说明	
表 6-5 取消呼叫过程说明	
表 6-6 拒绝呼叫过程说明	35
表 6-7 主动终止会话过程说明	
表 6-8 被动终止会话过程说明	
表 6-9 主叫号码显示说明	
表 6-10 发送DTMF过程说明	
表 6-11 发起视频呼叫过程说明	
表 6-12 接听视频呼叫过程说明	
表 6-13 取消呼叫过程说明	40
表 6-14 拒绝视频呼叫过程说明	40

表	6-15	主动终止视频会话过程说明	41
表	6-16	被动终止视频会话过程说明	42
表	6-17	呼叫前转过程说明	43
表	6-18	呼叫等待过程说明	43
表	6-19	呼叫保持过程说明	44
表	6-20	呼叫被保持过程说明	45
表	6-21	呼入阻止过程说明	45
表	6-22	盲转过程说明	46
表	6-23	咨询转过程说明	47
表	6-24	彩铃过程说明	50
表	6-25	基本呼叫过程说明	51
表	6-26	基本呼叫过程说明	52
表	6-27	拒绝呼叫过程说明	53
表	6-28	呼叫改向说明	54
表	6-29	呼叫保持过程说明	55
表	7-1 🖥	音视频录制和存储接口	58

图目录

冬	2-1	VoIP 客户端架构图	10
冬	3-1	License 激活过程	12
冬	3-2	License 解绑回收过程	13
冬	4-1	客户端初始化过程	15
冬	4-2	客户端销毁过程	16
冬	4-3	客户端启动过程	16
冬	4-4	客户端停止过程	17
冬	5-1	读取账号列表过程	20
冬	5-2	获取当前账号过程	21
冬	5-3	创建新账号过程	21
冬	5-4	删除账号过程	22
冬	5-5	用户打开过程	23
冬	5-6	用户关闭过程	23
冬	5-7	发起登陆过程	25
冬	5-8	取消登陆过程	26
冬	5-9	发起注销过程	27
冬	5-10)被动注销过程	28
冬	6-1	发起呼叫过程	32
冬	6-2	接听呼叫过程	33
冬	6-3	取消呼叫过程	34
冬	6-4	拒绝呼叫过程	35
冬	6-5	主动终止会话过程	35
冬	6-6	被动终止会话过程	36
冬	6-7	主叫号码显示过程	36

冬	6-8 发送DTMF过程	37
冬	6-9 发起视频呼叫过程	38
冬	6-10 接听视频呼叫过程	39
冬	6-11 取消呼叫过程	39
冬	6-12 拒绝视频呼叫过程	40
冬	6-13 主动终止视频会话过程	41
冬	6-14 被动终止视频会话过程	41
冬	6-15 呼叫前转过程	42
冬	6-16 呼叫等待过程	43
冬	6-17 呼叫保持过程	44
冬	6-18 呼叫被保持过程	45
冬	6-19 呼入阻止过程	45
冬	6-20 盲转过程	46
冬	6-21 咨询转过程	47
冬	6-22 彩铃过程	49
冬	6-23 基本呼叫过程	51
冬	6-24 取消呼叫过程	52
冬	6-25 拒绝呼叫过程	53
冬	6-26 呼叫改向过程	54
冬	6-27 呼叫保持过程	54
冬	6-28 呼叫转移盲转过程	55
冬	6-29呼叫转移咨询转过程	56

1. 产品概述

MTC (Multimedia Talk Client)是菊风公司的 VoIP 客户端框架,实现了 NGN/软交换/3GPP IMS VoIP 标准规定的客 户端业务功能,包括客户端登陆/注销、基本音视频通话、音视频补充业务等功能。MTC 的目的就是简化 GUI 客户端 对 VoIP 业务开发的复杂度,使得 GUI 开发人员可以轻装上阵,甚至不需要对 VoIP 业务标准有深入的了解。

1.1 用途

本手册的用途是帮助程序开发人员正确使用 VoIP Client 开发框架 MTC 来开发 VoIP 客户端软件产品。

1.2 阅读对象

本手册的读者应具备基本 C 语言编程知识、对 VoIP 业务标准 和 MTF 用户接口有基本了解。

1.3 内容简介

本手册详细介绍了 VoIP SDK (MTC) 的使用方法。

1.4 缩略词和缩写词

下表罗列了在本手册中使用的缩略词和缩写词。

缩略/缩写词	完整描述
ZOS	Zero Operation System
DNS	Domain Name System
MSF	Multi-Service Framework
MTF	Multimedia Telephone Framework
MTC	VoIP Client Framework
SIP	Session Initiation Protocol
URI	Uniform Resource Identifier
IMS	IP Multimedia Subsystem

表 1-1 缩略词和缩写词

1.5 参考

[1]《MTC VoIP Client API 参考指南》

2. 软件设计

2.1 客户端框架

菊风针对 VoIP 规范功能要求的客户端开发框架,称为 Multimedia Talk Client (MTC),用户可以通过 MTC 接口可以 实现符合 VoIP 标准的客户端。客户端框架如下图所示:



图 2-1 VoIP 客户端架构图

MTC 框架中包括以下功能模块:

- 登陆和注销
- 语音呼叫
- 补充业务(如呼叫等待等)
- 日志管理
- 业务参数管理

MTC 通过 MSF(Multi-Service Framework) 经由 SIP 协议栈实现与 VoIP 服务器的业务流程交互。

VoIP GUI 程序通过调用 MTC API 发起业务请求,例如登陆,发起语音呼叫等。 MTC 里面只是实现了基本的 GUI 界面操作之下的业务逻辑,使用 MTC 客户端的 GUI 集成开发会更加快速和简洁。

2.2 核心状态机

客户端在运行中分为以下状态:

- 未登陆状态
- 注册中状态
- 登陆成功状态

Juphcon

• 取消注册状态(注销后回退到未登陆状态)

从 MTC 的角度考虑,在不同状态下用户的操作范围是不一样的。比如登陆中状态不操作联系人列表、发起 VoIP 音视频会话等功能。登陆成功后可以操作所有功能。

MTC 的核心状态机是根据用户请求和 MTF 的事件状态来驱动整个 VoIP 业务逻辑。从客户端角度看,通过 MTC 业务请求接口和 GUI 回调机制以简便的方式实现业务功能。

GUI 客户端无须使用 MTC 的核心状态机接口,本节的目的是简单介绍 MTC 的业务逻辑实现过程。

2.3 用户数据管理

用户数据管理主要包括以下几个部分:

- 联系人管理
- 呼叫记录管理
- 短消息记录管理

联系人信息、呼叫记录和短消息记录等数据属于静态数据文件,保存在客户端本地

MTC 载入用户数据后,在内存中形成数据结构信息,GUI 可以根据数据的 ID 来引用数据内容

2.4 业务参数配置

MTC 中业务参数配置接口包括以下部分:

- 本地 IP 地址、日志基本等基本参数
- 用户信息
- SIP 本地和服务器参数
- 音视频媒体参数
- VoIP 的特殊业务参数

3. License 激活和回收

对于使用 License 控制的 VoIP SDK,需要调用相关接口实现 license 的激活和回收功能。

3.1 License激活和回收

3.1.1 接口和回调

接口名称	接口描述	GUI 回调
MtcLcs.Mtc_LcsActivate	根据 MtcCli.Mtc_CliInit 的返回值 , 判断是否发起 license 激活请求。	存在
MtcLcs.Mtc_LcsDeactivateDevice	当设备不使用时,删除 license ,并回收 license	存在

表 3-1 License 激活和回收接口

License 激活和回收需要界面(GUI)处理的回调:

用户操作	操作详细描述	
MtcLcs.Mtc_LcsActivate	回调函数 mtcLcsActivated 的 bActivated 参数为 true	License 激活请求成功。 需要在 该回调中重新调用 MtcCli.Mtc_CliInit 进行初始化
	回调函数 mtcLcsActivated 的 bActivated 参数为 false	License 激活请求失败
MtcLcs.Mtc_LcsDeactivateDevice	回调函数 mtcLcsDeviceDeactivatedM 的 bDeactivated 参数为 true	License 回收请求成功
	回调函数 mtcLcsDeviceDeactivatedM 的 bDeactivated 参数为 false	License 回收请求失败

表 3-2 License 回调处理

3.1.2 License 激活流程





2	使用 MtcCli.Mtc_CliInit 启动客户端系统(比如初始化 ZOS , XML 等)。
3	如 MtcCli.Mtc_CliInit 返 回 值 为 MtcLcsConstants.MTC_LCS_ERR_NEED_ACT_LICSEN, 则 调 用
<u> </u>	MtcLcs.Mtc_LcsActivate 发起 license 激活请求。
	由回调函数 mtcLcsActivated 上报 license 激活请求的结果。
	如果回调函数的参数 bActivated 为 true , 表示 license 请求成功 , 此时需要在 回调函数中重新
	发起 MtcCli.Mtc_CliInit 进行客户端的初始化。
	如果回调函数的参数 bActivated 为 false, 表示 license 请求失败 , 此时可由界面反馈错误结果。

表 3-3 客户端 License 激活过程说明

3.1.3 License 解绑流程



图 3-2 License 解绑回收过程



表 3-4 客户端解绑回收过程说明

3.1.4 License 激活和解绑示例代码

//设置 license 激活的回调

```
Mtc_LcsCbSetActivatedOk((PFN_MTCLCSACTIVATECB) OnLicenseActivateOk);
//回调函数实现
   void OnLicenseActivateOk(bool bActivated)
   {
       if(bActivated)
       {
           //license 激活成功,重新初始化客户端
           int iRet = Mtc_CliInit(CUtf8String(wzProfilePath));
           if(iRet = ZOK)
           {
               PostMessage(hwnd, UM_LAUNCHCLR, 0, 0);
               return;
           }
           else if(iRet == ZFAILED)
           {
               MessageBox(NULL,_T("Mtc_CliInit Error"), _T("ERROR"), MB_OK);
               exit(0);
           }
       }
       else
       {
           MessageBox(NULL,_T("License Activated Error"), _T("ERROR"), MB_OK);
           exit(0);
       }
     }
//发起 License 激活请求
    if(MTC_LCS_ERR_NEED_ACT_LICSEN == iRet)
   {
       Mtc_LcsCbSetActivatedOk((PFN_MTCLCSACTIVATECB)OnLicenseActivateOk);
       //向服务器 192.168.0.156 发起 license 请求并保存 license.sign 文件到目录 wzLicensePath
       Mtc_LcsActivate(CUtf8String("192.168.0.156"),
       CUtf8String("roytel"), ZNULL, CUtf8String(wzLicensePath));
       return FALSE;
   }
//解绑回收 license
    private void btnDeactivation_Click(object sender, RoutedEventArgs e)
    {
       string sLicensePath = System.AppDomain.CurrentDomain.BaseDirectory;
       sLicensePath = sLicensePath + "license.sign";
       //删除本地 license 并发起 license 解绑和回收请求
       int iRet = MtcLcs.Mtc_LcsDeactivateDevice("192.168.0.156", sLicensePath);
       if (MtcLcs.MTC_LCS_FILE_NOT_EXIST == iRet)
       {
           MessageBox.Show("License file not exist,\ns cannot deactivate the device!");
```

Juphcon

}

}

表 3-5 license 激活和回收示例代码

4. 客户端管理

客户端管理涉及一下过程处理:

- 销毁和初始化
- 启动和停止
- 参数设置

4.1 初始化和销毁

4.1.1 接口与回调

接口名称	接口描述	
Mtc_Cli.Mtc_CliInit	初始化 ZOS、XML 等资源,初始媒体模块,获取用户列表,启动业务引擎	
Mtc_Cli.Mtc_CliDestroy	关闭所有的业务、协议栈等任务,释放所有内存、媒体等资源	

表 4-1 客户端初始化和销毁接口

4.1.2 初始化操作过程



图 4-1 客户端初始化过程

1	设置一些预设参数		
2	设置 MTC 业务操作回调 (GUI Callback		
3	使用 Mtc_Cli.Mtc_CliInit 启动客户端系统(比如初始化 ZOS, XML 等)。 加载。	系统启动,用户信息等还没有	

表 4-2 客户端初始化过程说明

4.1.3 销毁操作过程



图 4-2 客户端销毁过程

 1 用户需要确保会话被管理、配置和日志信息已经保存好。一旦客户端停止,所有相关内存资源会进入无效 状态。
 2 调用 Mtc_Cli.Mtc_CliDestroy

表 4-3 客户端销毁过程说明

4.2 启动和停止

4.2.1 接口与回调

接口名称	接口描述
Mtc_Cli.Mtc_CliStart	启动 SIP 等协议栈任务、MTF 等业务组件、初始化 MTC 状态
Mtc_Cli.Mtc_CliStop	复位 MTC 业务状体、停止 MTF 等业务组件、停止 SIP 等协议栈任务
表 4-4 客户端启动和停止接口	

4.2.2 启动操作过程



图 4-3 客户端启动过程

1	在启动客户端前,需要确认用户账号已经被正确的打开(Mtc_Cli.Mtc_CliOpen)。原因是协议栈、尤其是	
· ·	业务框架有很多配置信息是跟账号绑定,如果账号没有被打开,则就没有这些配置参数,将会导致启动服	
	务无法正常使用。	
1	MTC 目前提供的本地 IP 地址的接口有 Mtc_CliDb.Mtc_CliDbGetLocallp,此接口只能返回一个本地 IP	
<u> </u>	地址。在实际部署中,PC 会存在多个网卡和 IP 地址,因此,建议本地 IP 地址采用操作系统提供的	
	接口直接获取和设置,在 Mtc_Cli.Mtc_CliStart 启动前用 Mtc_CliDb.Mtc_CliDbSetLocallp 设置选择的	
	本地 IP 地址。	
2	在启动前(Mtc_Cli.Mtc_CliStart),用户需要确保 MTF 等业务数据内存区中存放的本地 IP 地址是正确	
-	的地址。原因是 SIP 协议栈等任务启动时会根据本地 IP 地址绑定服务端口(比如 5060),如果地址不正	
	确,将导致绑定失败	

表 4-5 客户端启动过程说明

4.2.3 停止操作过程



在停止客户端前,需要确认用户账号已经被正确的关闭(Mtc_Cli.Mtc_CliClose)。
 用户使用 Mtc_Cli.Mtc_CliStop 停止客户端服务(协议及其业务)
 表 4-6 客户端停止过程说明

4.3 参数设置

客户端参数分为以下两种参数类型:

- 全局基本参数 (MTC 没有此方面的参数)
- 用户相关的网络、业务和媒体参数

客户端参数存放在每个用户的账号文件信息中,用户如果读取或修改参数,需要首先打开用户账号,否则获取的参数都是默认数值,在用户账号打开后会被覆盖掉。

4.3.1 音频设置

4.3.1.1 输入设备设置

接口名称	接口描述	
MtcMedia. Mtc_AudioGetInputDev	获取当前输入设备名称	
MtcMedia. Mtc_AudioSetInputDev	设置当前输入设备名称	
MtcMedia. Mtc_AudioGetInputDevCnt	获取所有输入设备数量	
MtcMedia. Mtc_AudioEnumInputDev	获取指定索引的输入设备名称,索引值范围是0至【设备数量-1】	
MtcMedia. Mtc_AudioSetInputVol	设置麦克风输入音量,范围0至100	
MtcMedia. Mtc_AudioGetInputVol	获取麦克风输入音量	
MtcMedia. Mtc_AudioMuteInput	设置麦克风静音状态	
表 4-7 音频输入设备接口		

4.3.1.2 输出设备设置

接口名称	接口描述
MtcMedia. Mtc_AudioGetOutputDev	获取当前输出设备名称
MtcMedia. Mtc_AudioSetOutputDev	设置当前输出设备名称
MtcMedia. Mtc_AudioGetOutputDevCnt	获取所有输出设备数量
MtcMedia. Mtc_AudioEnumOutputDev	获取指定索引的输出设备名称,索引值范围是0至【设备数量-1】
MtcMedia. Mtc_AudioSetOutputVol	设置扬声器输出音量,范围0至100
MtcMedia. Mtc_AudioGetOutputVol	
MtcMedia. Mtc_AudioMuteOutput	设置扬声器静音状态

4.4 样例代码

4.4.1 单帐号初始化

```
/* 初始配置文件目录为 profiles */
Mtc_Cli.Mtc_CliInit(...);
/* 定义业务对象类,继承 GUI 回调接口 */
MtcCallCb.mtcCallCbIncoming(...);
MtcCallCb.mtcCallCbAlerted(...);
MtcCallCb.mtcCallCbTalking(...);
MtcCallCb.mtcCallCbTermed(...);
/* 使用 profiles 目录下配置文件初始化模块 */
Mtc Cli.Mtc CliOpen(...);
/* 可以在此调用 db 设置接口设置参数 */
/* 启动 MTC 各个模块 */
if (Mtc_Cli.Mtc_CliStart() != MtcCommon.ZOK)
{
   Mtc_Cli.Mtc_CliDestory();
   Return MtcCommon.ZFAILED;
}
/* 其他初始化工作 */
/* 启动登录 */
Mtc_Cli.Mtc_CliLogin();
```

5. 用户管理

用户管理包括以下三部分:

- 账号管理,包括读取已有账号列表、创建新账号等操作
- 用户登录和注销,负责接入和取消电信运营商的服务网络
- 账号参数设置,包括本地网络地址、SIP 账号和服务器网络信息、音视频能力设置等

如果 GUI 开发者采用自有的账号管理,则读取、创建、删除、选择账号等操作接口无需考虑,但是打开和关闭用户 是必须要做的,目的是完成用户数据的载入(此时是默认值,需要重新设置)

5.1 账号管理

用户与账号在服务个体是相同的,但是名称上是有不同的,比如 Alice 在移动运营商获得的业务账号是 8000,本地 账号管理(如 Alice_CMCC)则是用户 Profile 的管理,所以,在 MTC 账号管理中,账号名称只是 Profile 的名称, 创建后就不能更改(除非删除)。所以在名称说法有以下约定:

- 账号名称:用户 Profile 的名称, MTC 中也是文件夹的名称, 如 Alice_CMCC。
- 用户名称: 服务提供商提供的业务账号, 如 8000。
- 用户昵称:用户个性化的显示名称,如 Alice Huang。

账号管理只是对账号目录进行管理,当新建或选择账号时,并不意味着账号的配置参数被载入。用户必须选择正确的 账号,通过打开账号操作才能完成账号的配置参数的载入,然后才能设置各种参数。

5.1.1 用户接口

接口名称	接口描述
MtcProf.Mtc_ProfGetUserSize	获取所有账号数目
MtcProf.Mtc_ProfGetUser	根据账号索引获取账号名称,索引范围是0到帐号数目-1
MtcProf.Mtc_ProfGetCurUser	获取当前账号
MtcProf.Mtc_ProfCreateUser	创建账号目录
MtcProf.Mtc_ProfDeleteUser	删除账号配置、日志文件和目录
MtcProf.Mtc_ProfExistUser	账号名称是否有效
MtcProf.Mtc_ProfSaveProvision	保存当前用户的配置文件
MtcProf.Mtc_ProfSaveContact	保存当前用户的联系人信息
MtcProf.Mtc_ProfSaveCallLogs	保存当前用户的呼叫历史记录
MtcProf.Mtc_ProfSaveImLogs	保存当前用户的 IM 历史记录
MtcProf.Mtc_ProfGetAbsoPath	获取绝对路径

表 5-1 用户管理接口

5.1.2 读取账号列表



在读取账号列表前,需要确保客户端资源已经成功的初始化(Mtc_Cli.Mtc_CliInit)。用户在初始化客户端时,还可以设置账号管理的目录(默认是 Profiles 目录)

1 使用 MtcProf.Mtc_ProfGetUserSize 获取账号目录中的所有账号(子目录)数目。

MtcProf.Mtc_ProfGetUser

2 使用 MtcProf.Mtc_ProfGetUser 获取索引位置的账号名称

表 5-2 读取账号列表说明

5.1.3 获取当前账号



图 5-2 获取当前账号过程

 在获取账号前,需要确保客户端资源已经成功的初始化 (Mtc_Cli.Mtc_CliInit)。

 使用 MtcProf.Mtc_ProfGetCurUser 获取当前账号。如果没有创建或选择过账号,则返回空、
 表 5-3 获取当前账号说明

5.1.4 创建新账号



图 5-3 创建新账号过程

	1	在创建账号前,需要确保客户端资源已经成功的初始化 (Mtc_Cli.Mtc_CliInit)。
(1	使用 MtcProf.Mtc_ProfExistUser 判断账号是否存在
(2	使用 MtcProf.Mtc_ProfCreateUser 创建新账号,MTC 会创建新文件夹

21

5.1.5 删除账号





5.2 打开与关闭

用户打开是指用户在选择账号后,把用户数据文件从账号目录中载入的过程。用户打开过程包括将各个业务和应用组件数据与 MSF 的数据组件实现绑定,然后加载用户账号中的数据文件内容到内存中。如果账号名称为空,则用户打开将只是初始化默认的数据内容。

用户关闭是指用户数据从内存中写入目录文件中,同时解除 MSF 数据组件与业务和应用组件的绑定。如果用户打开的时候账号为空,则写入操作将不被执行。

5.2.1 用户接口

Mtc Cli Mtc CliOpen 灯工对应账号夕秒的用户,我》用户粉提	らわれ おお 日本	接口描述	
	Cli.Mtc_CliOpen	打开对应账号名称的用户,载入用户数据	
Mtc_Cli.Mtc_CliClose 写入用户数据,并且关闭用户	Cli.Mtc_CliClose	写入用户数据,并且关闭用户	

表 5-6 打开和关闭接口

5.2.2 打开操作过程



1	在创建账号前,需要确保客户端资源已经成功的初始化(Mtc_Cli.Mtc_CliInit)。		
1	使用 MtcProf.Mtc_ProfExistUser 判断账号是否存在。如果在打开用户前已经有判断,则忽略此步骤。		
2	使用 Mtc_Cli.Mtc_CliOpen 打开用户,把用户数据载入内存中		

5.2.3 关闭操作过程





5.3 登陆与注销

客户端支持本地和服务器两种模式下登陆。本地登陆是指客户端无须向服务器发起注册,即可登陆客户端。但是,客 户端在商用部署时往往采用服务器模式下登陆。用户可以使用 Mtc_DbGetUserReg 接口判断登陆模式是否是服务器 模式。

5.3.1 接口与回调

用户实现登陆与注销的接口有:

接口名称	接口描述	GUI 回调存在
Mtc_Cli.Mtc_CliLogin	发起登陆请求	存在
Mtc_Cli.Mtc_CliLogout	发起注销请求	存在

表 5-9 登陆与注销用户接口

登陆与注销的接口相关的 GUI 回调处理:

用户操作	操作详细描述	
Mtc_Cli.Mtc_CliL ogin	MtcCliCb.mtcCliCbRegOk	SIP REGISTER 请求已经被服务器接受
	MtcCliCb.mtcCliCbRegFaile d	SIP REGISTER 请求已经被服务器拒绝或发送失败
	MtcCliCb.mtcCliCbLclLogin Ok	本地登陆成功
	MtcCliCb.mtcCliCbServLogi nOk	服务器登陆成功
	MtcCliCb.mtcCliCbLoginFail ed	服务器登陆失败
Mtc_Cli.Mtc_CliL ogout	MtcCliCb.mtcCliCbLclLogou t	本地注销成功
	MtcCliCb.mtcCliCbServLog out	服务器注销成功

表 5-10 登陆与注销 GUI 回调

5.3.2 发起登陆过程



图 5-7 发起登陆过程

1	用户账号已经成功打开(Mtc_Cli.Mtc_CliOpen)。
2	客户端已经成功启动 (Mtc_Cli.Mtc_CliStart)
1	使用 Mtc_Cli.Mtc_CliLogin 发起登陆请求。
2	获取登陆模式,如果是本地登陆,则设置好相关的 GUI 窗口布局,准备显示登陆成功窗口。
1	在客户端初始化的时候设置好 MtcCliCb.mtcCliCbServLoginOk , MtcCliCb.mtcCliCbLoginFailed GUI 回 调,登陆结果将会由这两个回调发起通知。
2	在 MtcCliCb.mtcCliCbServLoginOk 回调中显示登陆成功窗口。在 MtcCliCb.mtcCliCbLoginFailed 显示 登陆失败提示信息。
_	建议在登陆成功后,加载联系人列表、呼叫和短消息历史记录等本地用户数据。
	如果是本地登陆模式,在 Mtc_Cli.Mtc_CliLogin 函数里面会立即调用登陆通知 GUI 回调,因此对于建议 先调整好登陆成功的窗口布局,然后再发起登陆。在 Mtc_Cli.Mtc_CliLogin 函数返回后不建议做窗口调整 操作(除非此函数立即返回失败)
	如果是服务器模式,除非本地资源不足或发送失败,否则登陆结果会异步通知,Mtc_Cli.Mtc_CliLogin 会 立即返回。

表 5-11 发起登陆过程说明

5.3.3 取消登陆过程



图 5-8 取消登陆过程

1	客户端已经向服务器发起注册,处于正在登陆中。
1	使用 Mtc_Cli.Mtc_CliLogout 发起取消请求。
Δ	如果是本地登陆模式,可以立即准备登陆准备窗口。
1	在客户端初始化的时候设置好 MtcCliCb.mtcCliCbLclLogout , MtcCliCb.mtcCliCbServLogout GUI 回调,
	取消登陆结果将会由这两个回调发起通知。
2	在 MtcCliCb.mtcCliCbLclLogout 和 MtcCliCb.mtcCliCbServLogout 回调中处理取消登陆结果,显示对应
	的 GUI 窗口。
Δ	
	如果是本地登陆模式,在 Mtc_Cli.Mtc_CliLogout 函数里面会立即调用 MtcCliCb.mtcCliCbLclLogout 通知
	GUI 回调 ,
	如果是服务器模式,除非本地资源不足或发送失败(MtcCliCb.mtcCliCbLclLogout),否则取消登陆结果会异
	步通知(MtcCliCb.mtcCliCbServLogout), Mtc_Cli.Mtc_CliLogout 会立即返回。

表 5-12 取消登陆过程说明

5.3.4 发起注销过程



图 5-9 发起注销过程



5.3.5 被动注销过程



图 5-10 被动注销过程

客户端已经在服务器中成功注册,处于登陆成功状态。客户端必须要打开订阅注册状态的选项,接口为 Mtc_DbSetSubsReg。此后,在客户端被服务器注销的时候才会得到通知。

1 在客户端初始化的时候设置好 MtcCliCb.mtcCliCbServLogout GUI 回调,被动注销将会由此回调发起通知。在此回调中,会携带被注销的事件和具体参数。

事件包括:

	类型	说明	GUI 处理
	MtcCliConstants .MTC_CLI_REG_	注册已经超时	MtcCliCb.mtcCliCbServLogout
	ERR_TIMEOUT		回调中显示未登录状态
	MtcCliConstants .MTC_CLI.MTC_CLI_	注册信息无效	MtcCliCb.mtcCliCbServLogout
	REG_ERR_DEACTED		回调中显示未登录状态
	MtcCliConstants .MTC_CLI.MTC_CLI_	注册被拒绝 ,需要间隔一段	MtcCliCb.mtcCliCbServLogout
	REG_ERR_PROBATION	时间之后再发起注册	回调中显示未登录状态 并根据
			超时时间启动注册倒计时窗口,
			计时结束后 重新发起登录过程
1	在 MtcCliCb.mtcCliCbServLogout 回调中	, 使用 MtcCall.Mtc_SessTer	m 等接口关闭处于活动状态的音
	视频、短消息等会话。		
2	在 MtcCliCb.mtcCliCbServLogout 回调中	处理显示对应的 GUI 窗口。	

表 5-14 被动注销过程说明

6. 会话管理

MTC 会话管理支持以下业务:

- 语音呼叫
- 多路呼叫
- 呼叫保持

Juphcon

- 呼叫转移
- 呼叫前转
- 呼入阻止
- 媒体能力更新
- 发送 DTMF
- 主叫号码显示 Caller-ID
- 设置静音
- 设置音量

6.1 接口与回调

用户实现会话操作的接口有:

接口名称	接口描述	GUI 回调存在
MtcCall.Mtc_SessCall	发起会话呼叫	存在
MtcCall.Mtc_SessAnswer	应答会话呼入	存在
MtcCall.Mtc_SessTerm	终止会话	存在
MtcCall.Mtc_SessHold	保持会话	存在
MtcCall.Mtc_SessUnhold	取消保持会话	存在
MtcCall.Mtc_SessUTrsf	发起非协商会话转移	存在
MtcCall.Mtc_SessATrsf	发起协商会话转移	存在
MtcCall.Mtc_SessDiv	对呼入的会话进行呼叫前转	存在
MtcCall.Mtc_SessDtmf	发送 DTMF 信息	
MtcCall.Mtc_SessHasTalk	判断当前是否是通话状态	
MtcCall.Mtc_SessHasHold	判断当前是否处于呼叫保持状态	
MtcCall.Mtc_SessHasHeld	判断当前是否处于呼叫被保持状态	
MtcCall.Mtc_SessAudioLostRatio	获取音频丢包率	
MtcCall.Mtc_SessGetMicMute	获取麦克风静音状态	
MtcCall.Mtc_SessSetMicMute	设置麦克风静音状态	
MtcCall.Mtc_SessGetSpkMute	获取扬声器静音状态	
MtcCall.Mtc_SessSetSpkMute	设置扬声器静音状态	
MtcCall.Mtc_SessSetMixVoice	将会话加入本地混音	
MtcCall.Mtc_SessGetState	获取会话状态	
MtcCall.Mtc_SessGetPeerUri	获取会话对方的 URI , 从 From 获 取	

MtcCall.Mtc_SessGetPeerId	获取会话对方的 Caller-ID , 根据本 地设置从 PAI 和 From 获取	
MtcCall.Mtc_SessPeerIsFocus	判断对方是否是会议主持方	
MtcCall.Mtc_SessHasOfferAnswer	判断会话是否完成了媒体协商	
MtcCall.Mtc_SessGetEarlyMediaStatus	获取早期媒体的状态	

表 6-1 会话用户接口

会话接口相关的 GUI 回调处理:

用户操作	操作详细描述	
MtcCall.Mtc_SessC all	MtcCallCb.mtcCallCbOu tgoing	会话正在呼出中
	MtcCallCb.mtcCallCbAl erted	对方已经收到呼叫,处于振铃中
	MtcCallCb.mtcCallCbTal king	对方接受了呼叫请求,进入通话状态
	MtcCallCb.mtcCallCbTe rmed	对方拒绝或终止了呼叫请求
	MtcCallCb.mtcCallCbErr or	会话发生错误
*	MtcCallCb.mtcCallCbInc oming	收到远端的一个呼叫请求
MtcCall.Mtc_SessA nswer	MtcCallCb.mtcCallCbTal king	接受远端的呼叫请求,进入通话状态
*	MtcCallCb.mtcCallCbMd fyed	本方的媒体属性被修改
*	MtcCallCb.mtcCallCbRe fered	当前会话被对方转移
*	MtcCallCb.mtcCallCbRe direct	当前会话被对方前转
*	MtcCallCb.mtcCallCbRp led	当前会话被另一个会话代替
*	MtcCallCb.mtcCallCbErr or	当前会话处理发生错误
MtcCall.Mtc_SessH	MtcCallCb.mtcCallCbMd fyAcpt	对方接受修改媒体属性的请求

Juphcon

	MtcCallCb.mtcCallCbErr or	呼叫保持拒绝或处理错误
MtcCall.Mtc_SessU nhold	MtcCallCb.mtcCallCbMd fyAcpt	对方接受修改媒体属性的请求
	MtcCallCb.mtcCallCbErr or	呼叫保持解除拒绝或处理错误
MtcCall.Mtc_SessU pdate	MtcCallCb.mtcCallCbMd fyAcpt	对方接受修改媒体属性的请求
	MtcCallCb.mtcCallCbErr or	媒体能力更新拒绝或处理错误
MtcCall.Mtc_SessU Trsf	MtcCallCb.mtcCallCbTr sfAcpt	对方接受呼叫转移请求
	MtcCallCb.mtcCallCbTr sfTerm	呼叫转移终止
MtcCall.Mtc_SessA Trsf	MtcCallCb.mtcCallCbTr sfAcpt	对方接受呼叫转移请求
	MtcCallCb.mtcCallCbTr sfTerm	呼叫转移终止
	MtcCallCb.mtcCallCbErr or	取消视频通话能力拒绝或处理错误

表 6-2 会话 GUI 回调

6.2 语音呼叫

6.2.1 发起呼叫过程



图 6-1 发起呼叫过程



MtcCallCb.mtcCallCbAlerted 回调表明对方已经振铃或者网络侧播放铃音(包括多媒体彩铃)
 GUI 会话窗口更新会话状态(如 Ringing)。
 GUI 可以通过 MtcCall.Mtc_SessHasEarlyMedia 接口判断是否存在早期媒体,如果存在表明网络侧会播放铃音(彩铃)。
 MtcCallCb.mtcCallCbTalking 回调表明对方已经接听呼叫,语音流会双向接通。
 GUI 会话窗口更新会话状态(如 Talking)。
 表 6-3 发起呼叫过程说明

6.2.2 接听呼叫过程



图 6-2 接听呼叫过程



6.2.3 取消呼叫过程





1	客户端已经在服务器中成功注册,正处于登陆成功状态。
2	GUI 调用 MtcCall.Mtc_SessCall 发起呼叫。
1	MtcCallCb.mtcCallCbOutgoing 表明会话请求已经下发,等待会话响应
2	MtcCallCb.mtcCallCbAlerted 回调表明对方已经振铃或者网络侧播放铃音(包括多媒体彩铃)
1	GUI 调用 MtcCall.Mtc_SessTerm 取消呼叫,dwReason 是拒绝的原因,设置 MtcCliConstants.EN_MTC_CALL_TERM_REASON_NORMAL
4	与被叫拒绝呼入请求不同,主叫在取消发起的呼叫时并没有取消原因这一说法,dwReason 还要填值只是 出于接口一致的需要,事实上填成任何值都没问题
3	MtcCallCb.mtcCallCbTermed 表明有会话已经被终止
4	GUI 会话窗口更新会话状态(如 Ended),甚至关闭 GUI 会话窗口。
	表 6-5 取消呼叫过程说明

6.2.4 拒绝呼叫过程





1	客户端已经在服务器中成功注册,正处于登陆成功状态。
1	MtcCallCb.mtcCallCbIncoming 表明有会话邀请,等待用户拒绝
1	GUI 调用 MtcCall.Mtc_SessTerm 拒绝会话邀请。dwReason 是拒绝的原因,可以是
·	MtcCliConstants.EN_MTC_CALL_TERM_REASON_BUSY
	MtcCliConstants.EN_MTC_CALL_TERM_REASON_NORMAL。
	MtcCliConstants.EN_MTC_CALL_TERM_REASON_BUSY的意思是被叫正忙,服务器在收到此响应时可
	能会触发相应业务 , 如 CFB。此函数会立即返回。
-	表 6-6 拒绝呼叫过程说明

6.2.5 终止呼叫过程



图 6-5 主动终止会话过程

1	客户端已经在服务器中成功注册,且处于通话中状态。
1	GUI 调用 MtcCall.Mtc_SessTerm 终止会话。dwReason 设置为
	MtCCIIConstants.EN_MTC_CALL_TERM_REASON_NORMAL。
1	MtcCallCb.mtcCallCbTermed 表明有会话已经被终止

2) GUI 会话窗口更新会话状态(如 Ended), 甚至关闭 GUI 会话窗口。

表 6-7 主动终止会话过程说明



图 6-6 被动终止会话过程





MtcCallCb.mtcCallCbIncoming 表明有会话邀请,等待用户接听。
 ▲ GUI 可以使用 MtcCall.Mtc_SessGetPeerId 获取主叫号码显示(Caller-ID),同时此函数包括了对显示 限制的处理。MtcCall.Mtc_SessGetPeerUri 用于获取远端 URI (From UI)。用户可以根据此 2 种信息 查找对应的联系人是否存在,如果联系人存在号码,则还需要分析 URI 中的号码。
 2 GUI 会话窗口更新会话状态(如 Caller-ID)。

表 6-9 主叫号码显示说明

6.2.7 发送 DTMF 过程

DTMF (双音多频)是与呼叫服务器交互的语音频率的电话信号。可以选择 Auto 和 SIP Info 两种承载方式。



客户端已经在服务器中成功注册,登陆成功,正处于通话状态。
 GUI 调用 MtcCall.Mtc_SessDtmf 发送 DTMF 信号,调用 MtcRing.Mtc_RingPlay 播放 DTMF 声音。

表 6-10 发送 DTMF 过程说明

6.3 视频呼叫

6.3.1 发起呼叫过程



图 6-9 发起视频呼叫过程

1	客户端已经在服务器中成功注册,处于登陆成功状态。
1	GUI 调用 MtcCall.Mtc_SessCall 发起呼叫。设置输入参数中 bVoice 为 true , bVideo 为 true
	除非本地资源不足或发送失败,否则此函数会立即返回。
1	回调(通过 MtcCallCb.Mtc_CallCbSetOutgoing 设置) 表明会话请求已经下发,等待会话响应
2	回调(MtcCallCb.Mtc_CallCbSetAlerted)表明对方已经振铃或者网络侧播放铃音(包括多媒体彩铃)
0	GUI 会话窗口更新会话状态(如 Ringing)。
4	回调(MtcCallCb.Mtc_CallCbSetTalking)表明对方已经接听呼叫,视频流会双向接通。
5	GUI 会话窗口更新会话状态(如 Talking),视频显示窗口会显示视频图像。
	表 6-11 发起视频呼叫过程说明

6.3.2 接听呼叫过程





 Ŷep端已经在服务器中成功注册,处于登陆成功状态。

 回调(通过 MtcCallCb.Mtc_CallCbSetIncoming 设置)表明有会话邀请,等待用户接听

 GUI 在此回调中可以创建新的 GUI 会话窗口,显示会话状态(如 Call Incoming)。如果是视频呼叫,则 MtcCall.Mtc_SessHasVideo 会返回 true

 GUI 调用 MtcCall.Mtc_SessAnswer 接受会话邀请。设置输入参数中 bVoice 为 true, bVideo 为 true。 除非本地资源不足或发送失败,否则此函数会立即返回。

 GUI 会话窗口更新会话状态(如 Talking),使用 MtcCall.Mtc_SessStartVideo 启动本地显示图像,视频图像显示窗口会显示图像。

表 6-12 接听视频呼叫过程说明

6.3.3 取消呼叫过程





6.3.4 拒绝呼叫过程



图 6-12 拒绝视频呼叫过程

 客户端已经在服务器中成功注册,正处于登陆成功状态。
 回调(通过 MtcCallCb.Mtc_CallCbSetIncoming 设置)表明有会话邀请,等待用户拒绝。通过 MtcCall.Mtc_SessHasVideo的返回值(ZTRUE)来判断是否是视频呼叫。
 GUI 调用 MtcCall.Mtc_SessTerm 拒绝会话邀请。dwReason 是拒绝的原因,可以是 EN_MTC_CALL_TERM_REASON_NORMAL或EN_MTC_CALL_TERM_REASON_BUSY或 EN_MTC_CALL_TERM_REASON_DECLINE。EN_MTC_CALL_TERM_REASON_BUSY的意思是被叫 正忙,服务器在收到此响应时可能会触发相应业务,如CFB。此函数会立即返回。

表 6-14 拒绝视频呼叫过程说明

6.3.5 终止呼叫过程



图 6-13 主动终止视频会话过程

1	客户端已经在服务器中成功注册,且处于通话中状态。	
1	GUI 调用 MtcCall.Mtc_SessTerm 终止会话。dwReason 设置为	
	MtcCallConstants.EN_MTC_CALL_TERM_REASON_NORMAL 。此函数会立即返回。用户可以使用	
	MtcCall.Mtc_SessStopVideo 可以停止视频图像显示,甚至关闭视频图像显示窗口。	
1	回调(通过 MtcCallCb.Mtc_CallCbSetTermed 设置) 表明有会话已经被终止	
2	GUI 会话窗口更新会话状态(如 Ended),甚至关闭 GUI 会话窗口。	







1	客户端已经在服务器中成功注册,处于通	话中状态。	
1	回调(通过 MtcCallCb.Mtc_CallCbSetTermed 设置)表明有会话已经被对方终止。用户可以使用		
	MtcCall.Mtc_SessStopVideo 可以停止视	频图像显示,甚至关闭视频图像显示窗口。其中 dwReason 可	
	以是以下值:		
	EN_MTC_CALL_TERM_BYE	通话正常结束,收到 BYE 消息	

	EN_MTC_CALL_TERM_CANCEL	呼叫建立之前取消了呼叫,收到 CANCEL 消息
	EN_MTC_CALL_TERM_TIMEOUT	呼叫超时
	EN_MTC_CALL_TERM_BUSY	对方正忙,收到 486 消息
	EN_MTC_CALL_TERM_DECLINE	呼叫被拒绝,收到 603 消息
	EN_MTC_CALL_TERM_TRSFED	呼叫被转移
	EN_MTC_CALL_TERM_REDIRECT	呼叫被重定向
	EN_MTC_CALL_TERM_REPLACE	会话被替换
-		

2) GUI 会话窗口更新会话状态(如 Ended), 甚至关闭 GUI 会话窗口。

表 6-16 被动终止视频会话过程说明

6.4 呼叫前转

6.4.1 无条件前转

用户签约该业务后,前转由网络侧完成,终端不需要参与。

6.4.2 遇忙前转

用户签约该业务后,当终端已经达到最大呼叫数时,对于新呼入的呼叫终端响应 486,触发网络侧前转业务。GUI 不需要做处理。

6.4.3 无应答前转

用户签约该业务后,当用户在一定时间内对新呼入的呼叫无任何操作,则终端响应 480,触发网络侧前转业务。GUI 只需要参考接听呼叫流程显示界面,其他不需要做特殊处理。

6.4.4 不可及前转

用户签约该业务后,前转由网络侧完成,终端不需要参与。

6.4.5 呼叫改向



图 6-15 呼叫前转过程

客户端已经在服务器中成功注册,处于正处于登陆成功状态。

1 MtcCallCb.mtcCallCbIncoming 表明有会话邀请,等待用户阻止

1 GUI 调用 MtcCall.Mtc_SessDiv 把当前的呼入转移到另一个终端,此终端的 URI 由 pcUri 参数指定。此操作在服务器端会触发 CD 业务。

表 6-17 呼叫前转过程说明

6.4.6 未注册前转

用户签约该业务后,前转由网络侧完成,终端不需要参与。

6.5 呼叫等待



图 6-16 呼叫等待过程

	客户端已经有一路正在通话中的呼叫。
	MtcCallCb.mtcCallCbIncoming 表明有新的会话邀请,等待用户接听
Δ	GUI 在此回调中判断已经有一路正在通话的呼叫,则应该提示给用户一路新的呼叫正在等待应答。同时播
	放的提示声音应该与正常接听呼叫的声音有所区别。
1	GUI 调用 MtcCall.Mtc_SessAnswer 接受会话邀请。除非本地资源不足或发送失败,否则此函数会立即返
	回。
	如果用户决定接听新的一路呼叫,则 GUI 应该首先将前一路呼叫结束或者保持。所以建议,在呼叫等待提
Δ	示界面应有两个选项"结束当前通话,并接听"和"保持当前通话,并接听"。
2	GUI 会话窗口更新会话状态(如 Talking)。
	表 6-18 呼叫等待过程说明

6.6 呼叫保持



图 6-17 呼叫保持过程

1	客户端已经有一路正在通话中的呼叫。
1	GUI 调用 MtcCall.Mtc_SessHold 保持呼叫。
	呼叫保持需要信令交互,不能立刻完成。GUI应该显示正在保持呼叫,并限制用户呼叫保持成功之前,不能解除呼叫保持。
2	MtcCallCb.mtcCallCbHoldOk 指示呼叫保持操作成功,此时 GUI 应该将呼叫标记不同颜色或其他特征以 区别正常呼叫。
2	GUI 调用 MtcCall.Mtc_SessUnhold 解除保持呼叫。
3	呼叫保持需要信令交互,不能立刻完成。GUI应该显示正在解除保持呼叫,并限制用户在解除呼叫保持之前,不能呼叫保持。
4	MtcCallCb.mtcCallCbUnHoldOk 指示解除呼叫保持操作成功,此时 GUI 应该将呼叫标记成为正常呼叫。

表 6-19 呼叫保持过程说明



1	客户端已经有一路正在通话中的呼叫。
1 MtcCallCb.mtcCallCbHeld 指示呼叫被对方保持,GUI 应该将呼叫标记不同颜色或其他特征以区别	
	пЦ.
2	MtcCallCb.mtcCallCbUnHeld 指示对方已经解除呼叫保持,此时 GUI 应该将呼叫标记成为正常呼叫。
	表 6-20 呼叫被保持过程说明

6.7 呼叫限制

6.7.1 呼入阻止



图 6-19 呼入阻止过程

1	客户端已经在服务器中成功注册,处于正处于登陆成功状态。
1	MtcCallCb.mtcCallCbIncoming 表明有会话邀请,等待用户阻止
1	GUI调用 MtcCall.Mtc_SessTerm 拒绝会话邀请。dwReason 填成
·	MtcCliConstants.EN_MTC_CALL_TERM_REASON_DECLINE。
	MtcCliConstants.EN_MTC_CALL_TERM_REASON_DECLINE的意思是被叫动态阻止了当前的呼入,服
	务器在收到此响应时会触发 CB 业务。

表 6-21 呼入阻止过程说明

6.8 呼叫转移

6.8.1 盲转



图 6-20 盲转过程



6.8.2 咨询转



图 6-21 咨询转过程

1 7	客户端和用户 1 成功建立了会话,此会话处于保持中。
<mark>2</mark> ?	客户端和用户 2 成功建立了会话,此会话处于通话中。
1 0	GUI 调用 MtcCall.Mtc_SessATrsf 转移会话,dwTrsfSessId 参数指明了转移的目的地。当 dwTrsfSessId
톳	是客户端和用户1建立的会话 ID 时,表示客户端想把和用户2的会话转移成用户2和用户1之间的会话。
4	转移业务成功完成时,客户端和用户1和用户2的所有会话都会终止,用户2和用户1处于通话中。
1	MtcCallCb.mtcCallCbTrsfAcpt 表明呼叫转移业务已被服务器接受
人	被服务器接受并不表示转移已经成功,接受的意思是服务器通过检查策略接受了呼叫转移的业务请求。
2 1	MtcCallCb.mtcCallCbTermed 表明和用户 1 的会话已经被终止
3 (GUI 会话窗口更新会话状态(如 Ended),甚至关闭 GUI 会话窗口。
4 ►	MtcCallCb.mtcCallCbTermed 表明和用户 2 的会话已经被终止
5 (GUI 会话窗口更新会话状态(如 Ended),甚至关闭 GUI 会话窗口。
<u>6</u> N	MtcCallCb.mtcCallCbTrsfTerm 表明呼叫转移业务已经结束
	由于网络的原因,MtcCallCb.mtcCallCbTrsfTerm 和两个 MtcCallCb.mtcCallCbTermed 之间的先后顺序有
F	可能会不一样,GUI要考虑到这种可能性。

表 6-23 咨询转过程说明

6.9 铃音播放

6.9.1 本地铃音

本地铃音主要分为两类:按键铃音和事件铃音。

类型	描述		接口
按键铃音	用户按下拨号盘上的数字和星、井按键时播放的声音,其	类型定义在	MtcRing.Mtc_RingPlay
	MtcCliConstants.EN_MTC_TONE_TYPE。		
	铃音播放应该在 GUI 界面对应拨号盘对应按键处理中调用	∃。	
事件铃音	发生某一事件时播放的声音,现在提供的声音类型定义在 MtcCliConstants.EN_MTC_RING_TYPE。此部分铃音播放 内部处理直接调用。如果需要 GUI 处理,可以修改 MTC p 具体类型包括	牧是在 MTC 内部实现。	MtcRing.Mtc_RingPlay
	类型	说明	
	MtcCliConstants. EN_MTC_RING_RING	收到语	
		音呼叫	
	MtcCliConstants. EN_MTC_RING_RING_BACK	发起呼	
		пц	
	MtcCliConstants. EN_MTC_RING_TERM	呼叫结	
		束	
	MtcCliConstants. EN_MTC_RING_CALL_FAILED	呼叫失 败	

6.9.2 早期媒体

早期媒体是指在建立通话之前,呼叫发起方会收到来自网络侧的媒体数据,一般为彩铃或者多媒体彩铃。

- 彩铃:只有声音媒体
- 多媒体彩铃:包括声音和视频媒体

6.9.2.1 彩铃



图 6-22 彩铃过程

彩铃业务的过程与一般发起语音呼叫的过程基本一致,差别在于收到 18x 响应中,是否带有早期媒体的 SDP 协商参数。如果没有,则 MTC 则会播放本地铃音。如果有,则会播放从网络侧收到的早期媒体数据。此业务实现,不需要GUI 做特殊处理。具体过程如下:



一般情况下,GUI会话窗口可忽略此状态,但是如果会话窗口中的呼出状态由 MtcCallCb.mtcCallCbOutgoing 驱动的话,可以在回调中找到会话窗口,然后显示呼出状体(如 Calling)

2) MtcCallCb.mtcCallCbAlerted 回调表明对方已经振铃,即收到 18x 响应。

2) 如果响应中携带早期媒体的 SDP 参数 , 则 MTC 会播放网络侧收到的早期媒体数据 , 即彩铃。

GUI 会话窗口更新会话状态 (如 Ringing)。

- GUI 可以通过 MtcCall.Mtc_SessGetEarlyMediaStatus 接口判断是否存在早期媒体,如果存在表明网络侧会播放铃音(彩铃)。
- 4 MtcCallCb.mtcCallCbTalking 回调表明对方已经接听呼叫,语音流会双向接通。
 - 🕨 GUI 会话窗口更新会话状态(如 Talking)。

表 6-24 彩铃过程说明

6.10 会话定时器

MTC 默认支持会话定时器功能, GUI 不需要有对应处理。

6.11 拨号 URI

GUI 调用 MtcCall.Mtc_SessCall 发起语音或视频呼叫。其中 URI 的参数需要 GUI 根据终端本地策略基于 MtcUri .Mtc_UriFormat 等接口组装成为 SIP URI 或者 TEL URI。

6.11.1 SIP URI

6.11.1.1 本地登陆模式组装

如果是有效的 IP 地址 建议组装为 sip:unknown@ip_address 例如 sip:unknown@192.168.0.78 其中 192.168.0.78 为用户输入内容。

如果是有效的域名 ,建议组装为 sip:unknown@domain ,例如 sip:unknown@sip.juphoon.com ,其中 sip.juphoon.com 为用户输入内容。

6.11.1.2 服务器登陆模式

直接封装为 sip:phone@domain,例如 sip:13486651032@sip.juphoon.com,其中 13486651032为用户输入内容, sip.juphoon.com 为用户设置的域名。

6.11.2 TEL URI

直接封装为 tel:phone,例如 tel:13486651032,其中 13486651032 为用户输入内容。

6.11.3 模拟 POTS 终端业务定制

GUI 不需要特殊处理,只需要根据本地策略,将用户输入组装成 SIP URI 或 TEL URI,并发起呼叫即可。

6.12 会话举例

6.12.1 基本呼叫过程



图 6-23 基本呼叫过程

1	主叫方调用 MtcCall.Mtc_SessCall 发起呼叫。输入参数中,bVoice 设置为 ZTRUE 表示加入音频,bVideo
	为 ZIROE 表示加入视频。 pcon 表示想要呼叫的对方 ORI。 pcon , dwcookie , pdwsessid 参数说明见 521
	<u>U.2</u> 0
2	MtcCallCb.mtcCallCbOutgoing 回调表明发起会话请求已经下发,等待会话响应。
	MtcCallCb.mtcCallCbIncoming 回调表明有会话邀请,等待用户接听。
3	MtcCallCb.mtcCallCbAlerted 回调表明对方已经振铃。
2	被叫方调用 MtcCall.Mtc_SessAnswer 接受会话邀请。输入参数中,bVoice 设置为 ZTRUE 表示加入音频,
-	bVideo 为 ZTRUE 表示需要加入视频。pcUri 表示想要呼叫的对方 URI。pcUri,dwCookie,pdwSessId
	参数况明见 <u>5.2.1</u> 。
3	MtcCallCb.mtcCallCbTalking 回调表明接听呼叫,语音流会双向接通。双方建立会话成功。
4	MtcCallCb.mtcCallCbTalking 回调表明对方已经接听呼叫,语音流会双向接通。双方建立会话成功。
6	主叫方调用 MtcCall.Mtc_SessTerm 终止会话。dwReason 表示终止会话的原因,在终止会话时应该设置
	为 MtcCliConstants.EN_MTC_CALL_TERM_REASON_NORMAL 。其他业务终止会话原因详见 5.2.5 终
	山原因的央义拥处。
4	被叫方 MtcCallCb.mtcCallCbTermed 回调表明此会话已经被终止。
6	主叫方 MtcCallCb.mtcCallCbTermed 回调表明终结会话请求已经下发,此会话已经被终止。

表 6-25 基本呼叫过程说明

6.12.2 取消呼叫过程



图 6-24 取消呼叫过程

1	主叫方调用 MtcCall.Mtc_SessCall 发起呼叫。 输入参数中, bVoice 设置为 ZTRUE 表示加入音频, bVideo
<u> </u>	为 ZTRUE 表示需要加入视频。pcUri 表示你想要呼叫的对方 URI。pcUri, dwCookie, pdwSessId 参数说
	明见 <u>5.2.1</u> 。
2	MtcCallCb.mtcCallCbOutgoing 回调表明会话请求已经下发,等待会话响应。
1	MtcCallCb.mtcCallCbIncoming 回调表明有会话邀请,等待用户接听。
3	MtcCallCb.mtcCallCbAlerted 回调表明对方已经振铃或者网络侧播放铃音(包括多媒体彩铃)
4	主叫方调用 MtcCall.Mtc_SessTerm 终止会话。dwReason 设置成
-	MtcCliConstants.EN_MTC_CALL_TERM_REASON_NORMAL。与被叫拒绝呼入请求不同,主叫在取消
	发起的呼叫时并没有取消原因这一说法,dwReason 还要填值只是出于接口一致的需要,事实上填成任何
	值都没问题。其他业务终止会话原因详见 5.2.5 终止原因的英文描述。
5	主叫方 MtcCallCb.mtcCallCbTermed 回调表明取消会话请求已经下发,此会话被终止。
2	被叫方 MtcCallCb.mtcCallCbTermed 回调表明此会话已经被终止。

表 6-26 基本呼叫过程说明

6.12.3 拒绝呼叫过程



图 6-25 拒绝呼叫过程



表 6-27 拒绝呼叫过程说明

6.12.4 呼叫改向过程



图 6-26 呼叫改向过程

- 1 主叫方调用 MtcCall.Mtc_SessCall 发起呼叫。 输入参数中, bVoice 设置为 ZTRUE 表示加入音频, bVideo 为 ZTRUE 表示需要加入视频。
- 1 被叫 MtcCallCb.mtcCallCbIncoming 回调表明有会话邀请,等待用户接听。
- 2 MtcCallCb.mtcCallCbAlerted 回调表明对方已经振铃。
- 2) 被叫方 GUI 调用 MtcCall.Mtc_SessDiv 把当前的呼入转移到另一个终端 此终端的 URI 由 pcUri 参数指定。
- 3 MtcCallCb.mtcCallCbRedirect回调表明会话被转向到另外一个终端。GUI应相应调整界面显示。
- 3 被叫方 MtcCallCb.mtcCallCbTermed 回调表明此会话已经被终止

表 6-28 呼叫改向说明

6.12.5 呼叫保持过程



图 6-27 呼叫保持过程

1	发起保持方 GUI 调用 MtcCall.Mtc_SessHold 保持呼叫。
1	被保持方 MtcCallCb.mtcCallCbHeld 回调指示呼叫被对方保持
2	MtcCallCb.mtcCallCbHoldOk 指示呼叫保持操作成功。
3	发起保持方调用 MtcCall.Mtc_SessUnhold 解除保持呼叫。
2	MtcCallCb.mtcCallCbUnHeld 指示对方已经解除呼叫保持。
4	MtcCallCb.mtcCallCbUnHoldOk 指示解除呼叫保持操作成功。

表 6-29 呼叫保持过程说明

6.12.6 呼叫转移过程

6.12.6.1 盲转过程



图 6-28 呼叫转移盲转过程

1	GUI 调用 MtcCall.Mtc_SessUTrsf 转移会话,pcUri 参数指明了转移的目的地,如用户 2 的 URI。
2	MtcCallCb.mtcCallCbTrsfAcpt 表明呼叫转移业务已被服务器接受。被服务器接受并不表示转移已经成功,
-	接受的意思是服务器通过检查策略接受了呼叫转移的业务请求。
3	MtcCallCb.mtcCallCbTermed 回调表明会话已经被终止。
4	MtcCallCb.mtcCallCbTrsfTerm 表明呼叫转移业务已经结束。由于网络的原因,
	MtcCallCb.mtcCallCbTrsfTerm 可能发生在 MtcCallCb.mtcCallCbTermed 之前。
	表 6-42 呼叫转移盲转过程说明

6.12.6.2 咨询转过程



图 6-29 呼叫转移咨询转过程

6	GUI 调用 MtcCall.Mtc_SessATrsf 转移会话,dwTrsfSessId 参数指明了转移的目的地。当 dwTrsfSessId		
-	是客户端和用户1建立的会话 ID 时,表示客户端想把和用户2的会话转移成用户2和用户1之间的会话。		
	转移业务成功完成时,客户端和用户1和用户2的所有会话都会终止,用户2和用户1处于通话中。		
2	MtcCallCb.mtcCallCbTrsfAcpt 表明呼叫转移业务已被服务器接受。被服务器接受并不表示转移已经成功,		
-	接受的意思是服务器通过检查策略接受了呼叫转移的业务请求。		
3	MtcCallCb.mtcCallCbTermed 表明和用户 1 的会话已经被终止。		
4	MtcCallCb.mtcCallCbTermed 表明和用户 2 的会话已经被终止。		
6	MtcCallCb.mtcCallCbTrsfTerm 表明呼叫转移业务已经结束。由于网络的原因,		
	MtcCallCb.mtcCallCbTrsfTerm 和两个 MtcCallCb.mtcCallCbTermed 之间的先后顺序有可能会不一样。		

表 6-43 呼叫转移咨询转说明

7. 媒体录制存储

7.1 音视频录制存储接口

7.1.1 接口

接口名称	接口描述
MtcCall. Mtc_SessRecMicStart	录制麦克风声音。
	参数说明:
	pcFileName:录制内容保存的文件名称。如: "c:\documetns\a.wav"。
	ucFileType:录制的文件编码类型。类型定义参考: EN_MTC_MFILE_TYPE。
	如:EN_MTC_MFILE_TYPE.EN_MTC_MFILE_WAV_PCMU
MtcCall. Mtc_SessRecMicStop	停止录制麦克风声音
MtcCall .Mtc_SessRecPlayStart	录制某路通话的扬声器声音。
	参数说明:
	dwSessId: 某路通话的会话 ID 。如果 ID 为非法值,则录制混
	音。
	pcFileName: 文件名。同上。
	ucFileType:文件编码类型。同上。
MtcCall.Mtc_SessRecPlayStop	停止录制某路通话的扬声器声音。
	参数说明:
	dwSessld:某路通话的会话 ID 。如果 ID 为非法值 ,则停止录制
	混音。
MtcCall. Mtc_SessRecCallStart	录制整个会话的声音(包括麦克风和扬声器的声音)。
	参数说明:
	pcFileName: 文件名。同上。
	ucFileType:文件编码类型。同上。
MtcCall. Mtc_SessRecCallStop	停止录制整个会话声音。
MtcCall.Mtc_SessRecRecvVideoStart	录制收到的视频数据。
	参数说明:
	dwSessId:录制的会话 ID。
	pcFileName:录制文件文件名。同上。
	ucFileType:文件类型。同上。
	iWidth:视频的宽度(像素)

	iHeight:视频高度(像素)
	bWithAudio:是否录制音频数据
MtcCall. Mtc_SessRecRecvVideoStop	停止录制收到的视频数据。
	参数说明:
	dwSessId:录制的会话 ID。
MtcCall.Mtc_SessRecSendVideoStart	录制发送的视频数据。
	参数说明:
	dwSessId:录制的会话 ID。
	pcFileName:录制文件文件名。同上。
	ucFileType:文件类型。同上。
	iWidth:视频的宽度(像素)
	iHeight:视频高度(像素)
	bWithAudio:是否录制音频数据
MtcCall.Mtc_SessRecSendVideoStop	停止录制发送的视频数据。
	参数说明:
	dwSessId:录制的会话 ID。
MtcCall. Mtc_SessRenderSnapshot	远端视频截屏。
	参数说明:
	dwSessId:录制的会话 ID。
	pcFileName:录制文件文件名。同上。
	ucFileType:文件类型。参考 EN_MTC_CALL_IMG_TYPE。如:
	EN_MTC_CALL_IMG_TYPE.EN_MTC_CALL_IMG_JPEG
MtcCall. Mtc_SessCaptureSnapshot	本地采集视频截屏。
	参数说明:
	dwSessId:录制的会话 ID。
	pcFileName:录制文件文件名。同上。
	ucFileType:文件类型。参考 EN_MTC_CALL_IMG_TYPE。如:
	EN_MTC_CALL_IMG_TYPE.EN_MTC_CALL_IMG_JPEG

表 7-1 音视频录制和存储接口