

提示JTAPI问题和案例研究

TAC

文档ID116060

已更新：九月06，2013

贡献用迈克尔Mendoza和Bakthavatsal Muralidharan，Cisco TAC工程师。



[下载 pdf文档](#)



[打印](#)

[反馈](#)

相关产品

- [Cisco Unity Express](#)

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[提示与CUCM概述的JTAPI集成](#)

[高层次呼叫流示例](#)

[跟踪的能力提升计划和集](#)

[实时JTAPI CCN跟踪](#)

[JTAPI CCN Trace日志](#)

[收集Trace日志文件](#)

[必须有详细信息，在您检查日志前](#)

[基本CTI概念](#)

[普通的CTI呼叫状态](#)

[什么Trace记录应该看起来象](#)

[CTI RP和波尔特注册](#)

[基本呼叫转发对语音邮件](#)

[新的呼叫和重定向对可用端口](#)

[新的呼叫和重定向对可用端口](#)

[对CTI端口的新的呼叫](#)

[CTI端口接受重定向呼叫](#)

[媒体协商](#)

[呼叫断开](#)

[MWI开/关信令](#)

[提示开MWI闪亮指示线路的3001](#)

[删除从邮箱的消息的已拨号DTMF编号'3'](#)

[提示为线路3001关闭MWI闪亮指示](#)

[实时CCN日志](#)

[呼叫建立](#)

[呼叫断开](#)

[排除故障案例研究](#)

[连接问题](#)

[用户没CTI启用](#)

[CUCM CTI Manager服务发生故障](#)

[配置不匹配](#)

[CUCM呼叫路由问题](#)

[波尔特注册故障排除的清单](#)

[相关信息](#)

[相关的思科支持社区讨论](#)

简介

本文提供信息关于怎样排除故障Cisco Unity Express (CUE) Java电话应用可编程接口(JTAPI)。另外，本文提供信息和on命令如何启用，收集和查看不同的跟踪和日志与故障排除案例研究示例。

先决条件

要求

Cisco 建议您了解以下主题：

- 基础知识如何通过Web管理接口配置和使用Cisco Unified Communications Manager (CUCM)。
- 基本熟悉用计算机电话接口(CTI)端口和路由点(RP)在CUCM。
- 与Cisco Unity Express命令行界面的基本熟悉。

使用的组件

本文档中的信息基于以下软件和硬件版本：

- Cisco Unity Express版本3.x或以上。
- Cisco Unified Communications Manager版本7.x或以上。

集成使用的方法为与Cisco Unified Communications Manager的Cisco Unity Express仅适用;不与Cisco Unified Communications Manager Express (CUCME)。

必须为CUCM准许Cisco Unity Express，不是CUCME。提示可能在任何时间集成与CUCM或CUCME和相应地准许。

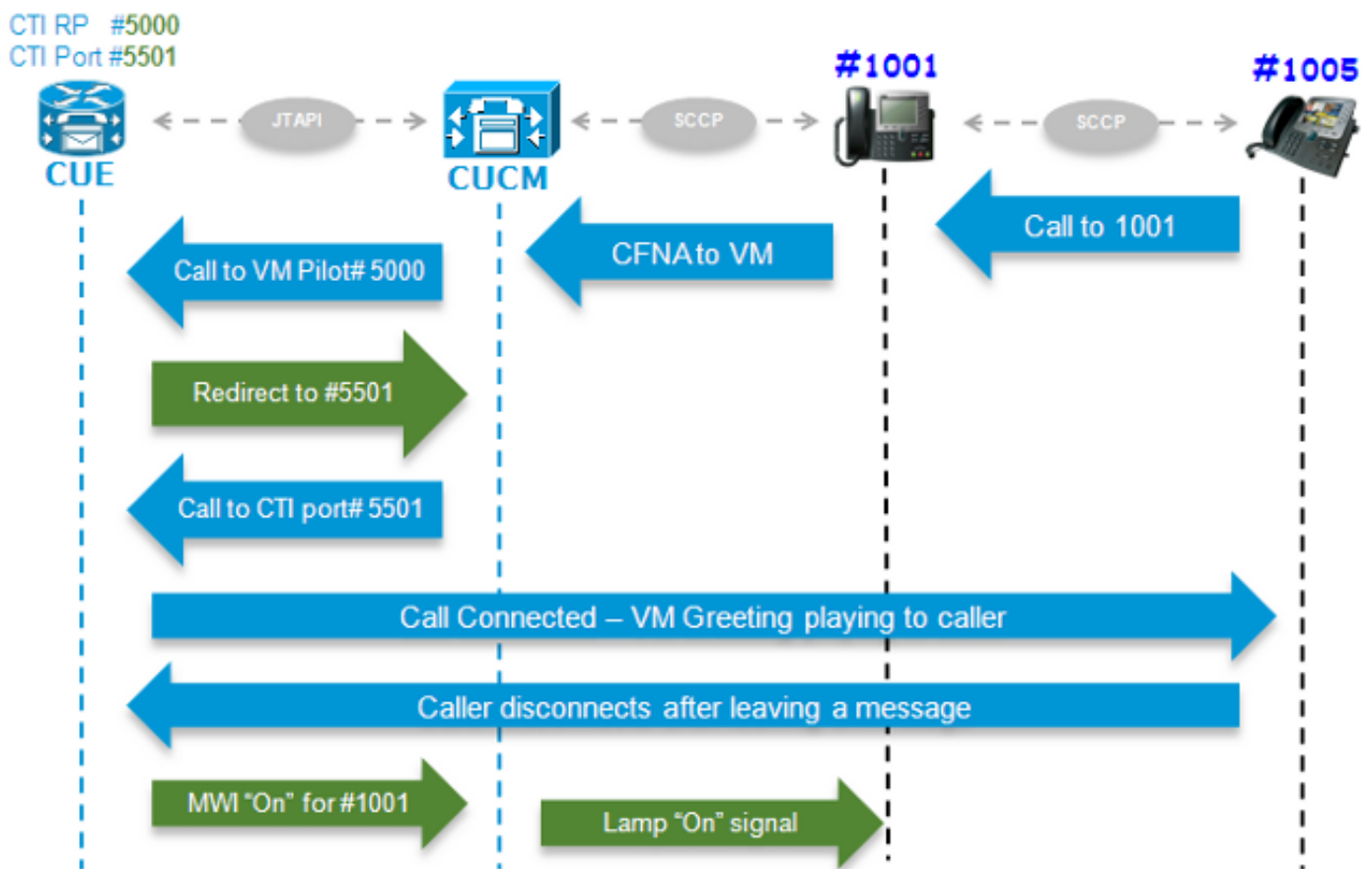
本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

提示与CUCM概述的JTAPI集成

通过语音邮件(VM)是可能的和自动话务员(AA)功能的JTAPI协议集成与CUCM的提示。推荐此解决方案，当您设置VM功能和基本AA呼叫处理一个的时或并联支路站点有很小数量的用户的注册到CUCM服务器。这不要一个发展完善的Cisco Unity语音邮件服务器，然而一个更加价格合理的实施。同时，提示为其分组也提供生存能力选项并且故障切换对会话启动的协议(SIP)，当对CUCM的连接丢失时。

提示能向CUCM登记通过JTAPI并且控制CTI路由点和CTI端口。这允许您控制和管理提示作为一个另外的终端通过CUCM，以及实现配置和交互作用与其他终端在集群。

高层次呼叫流示例



有目录号(DN)的1005最终用户告诉有DN的1001用户。呼叫在一些秒钟之后转发，如果呼叫没有应答，呼叫转发无应答(CFNA)，对在用户1001 VM配置文件配置的VM编号。CUCM然后发送呼叫对已配置的VM试验5000，指向与DN 5000的CTI RP是由提示控制的。提示VM应用程序被触发，并且呼叫通过对一个可用的CTI端口(DN 5501)的JTAPI重定向媒体组织的。音频问候语作用和用户能留下消息或与系统呼应通过Dual Tone Multi-frequency (DTMF)音。当呼叫方结束呼叫时，提示发信号CUCM设置分机的1001消息等待指示符(MWI)闪亮指示为"on"通过JTAPI。CUCM然后传送小型客户机控制协议(SCCP)信息开在电话的灯，以及显示在显示的信封征兆，因此用户1001知道有在邮箱的一个新的VM消息。

跟踪的能力提升计划和集

有跟踪的两种类型：

- 实时JTAPI思科通信网络(CCN)跟踪
- JTAPI CCN trace日志

实时JTAPI CCN跟踪

- 实时JTAPI CCN跟踪。(启用这些跟踪不要求提示模块的重新加载。)
- 输出不是一样广泛的象CCN trace日志，但是他们不是情报的。

输入这些命令为了启用跟踪：

```
no trace all
trace ccn SubsystemJtapi all
```

输入此命令为了验证他们启用：

```
CUE# show trace
MODULE ENTITY SETTING
ccn SubsystemJtapi ffffffff
```

输入此命令为了收集输出：

```
CUE# show trace buffer ?
containing Only display events matching a regex pattern
long Show long format
short Show short format
tail Wait for events and print them as they occur !!
```

输入CTRL-C终止实时记录日志到控制台。

JTAPI CCN Trace日志

在JTAPI CCN trace日志启用为了日志能变得填充后，提示模块的重新加载要求。这些日志，**messages.log**和**atrace.log**，可以非常详细或隐秘以及更加情报和详细。有四本不同的日志：

- **atrace.log**

默认情况下启用在网络模块(NMs)默认情况下，但是禁用为先进的集成模块(目标)。输入**log trace local enable**命令为了启用。它记载对10 Mb本地或对FTP服务器。为了重新启动日志，请输入**log trace local disable**命令或**no log trace local enable**命令;然后请输入**log trace local enable**命令。输入**清楚跟踪文件**命令为了清除**atrace.log**。必须由技术支持中心(TAC)解码数据。

- **messages.log**

这些是包含系统消息，例如资讯台、亚里桑，错误和致命的日志。

- **CiscoJtapi1.log**和**CiscoJtapi2.log**

他们记录所有JTAPI相关信令和事件。这些日志是更加容易了解和非常情报。当**CiscoJtapi1.log**变得全双工和反过来也是一样时地，**CiscoJtapi2.log**开始填充。

不管哪些跟踪设置，系统恢复对默认跟踪级别在重新加载以后。为了更改这些默认设置，以便他们生存重新启动，您必须输入**log trace boot**命令。这是命令启用他们：

```
CUE#(CONFIG)> log console info !!
ccn trace jtapi deb all
```

```
ccn trace jtapi info all
ccn trace jtapi warn all
log trace boot
reload
```

输入此命令为了验证他们启用：

```
CUE# show ccn trace jtapi
Warning: 1
Informational: 1
Jtapi Debugging: 1
Jtapi Implementation: 1
CTI Debugging: 1
CTI Implementation: 1
Protocol Debugging: 1
Misc Debugging: 1
```

这是查看日志的步骤：

1. 输入**show logs**命令为了查看在提示存储的日志文件的列表。
2. **.prev**文件扩展意味着这是一个更旧的跟踪文件而不是当前活动日志文件的备份。
3. 您能解压缩他们到一个外部FTP服务器。
4. 您能也查看被记录到这些文件实时从提示的终端监视器消息的输出。

收集Trace日志文件

解压缩日志对外部FTP用这些命令：

```
CUE# show ccn trace jtapi
Warning: 1
Informational: 1
Jtapi Debugging: 1
Jtapi Implementation: 1
CTI Debugging: 1
CTI Implementation: 1
Protocol Debugging: 1
Misc Debugging: 1
```

显示日志对提示终端监视器用**show log名称<logname>**命令。示例如下：

```
CUE# show log name messages.log ?
containing Only display events matching a regex pattern
paged Display in page mode
tail Wait for events and print them as they occur
<cr>
```

atrace.log编码;因此您能不仅显示它实时用**show log name**命令。

必须有详细信息，在您检查日志前

您必须从与您排除故障的问题的呼叫得到，至少，此中被概述的所有详细信息，以便您能容易地跟踪和了解跟踪：

- 主叫号码
- 被叫号码
- 重定向号码
- CTI RP DN和设备名
- CTI端口号和设备名

- JTAPI用户
- 时间范围呼叫发生了

基本CTI概念

供应商：CTI服务的服务商。应用程序通过打开供应商建立一个CTI会话。

用户：应用程序关联与用户。

设备：注册对CUCM的设备。

线路：在CTI支持的设备的DN外观。

呼叫ID (callLegID)：关联用在呼叫的一个呼叫段。

全局呼叫(callid)：识别单个呼叫的所有呼叫段。

普通的CTI呼叫状态

```
CUE# show log name messages.log ?
containing Only display events matching a regex pattern
paged Display in page mode
tail Wait for events and print them as they occur
<cr>
```

什么Trace记录应该看起来象

在您能查找不正确信令前，您首先需要了解什么此信令将看起来象下面正常操作;如此，当他们正常时，工作此部分显示信令的片断输出您在不同情况下将看到。

并且请注意从这些日志的所有信令汇总显示仅相关详细信息，因为他们包含非常是相当繁琐和重复性的详细信息。

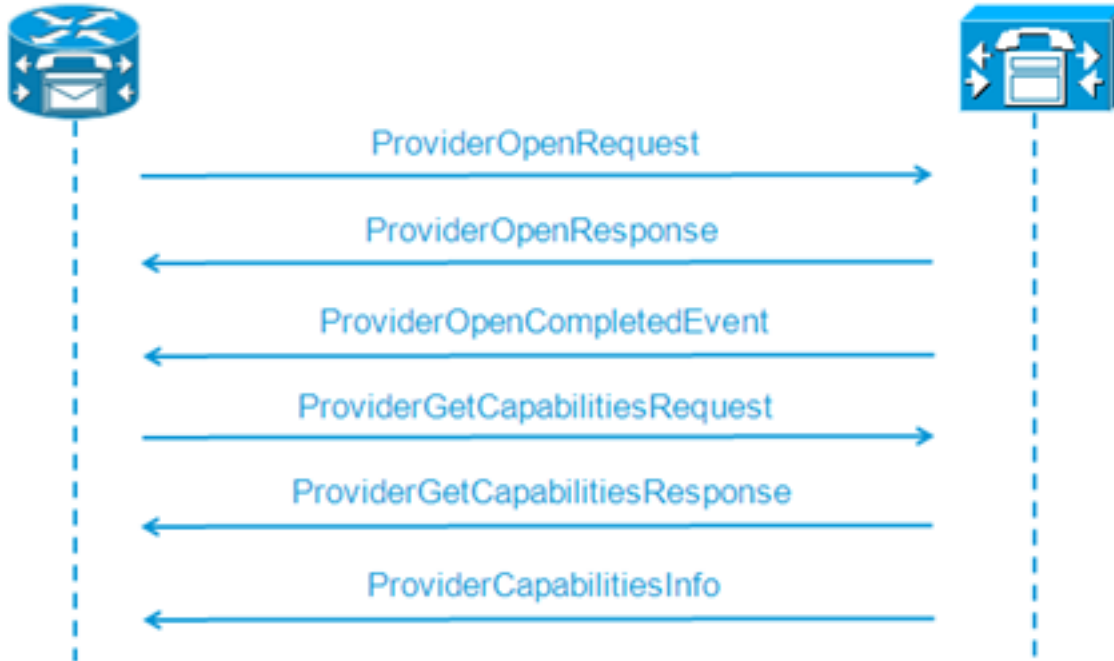
这是使用的配置的详细信息：

```
CUE# show log name messages.log ?
containing Only display events matching a regex pattern
paged Display in page mode
tail Wait for events and print them as they occur
<cr>
```

CTI RP和波尔特注册

(从CiscoJtapi1 /Cisco Jtapi2日志的输出)

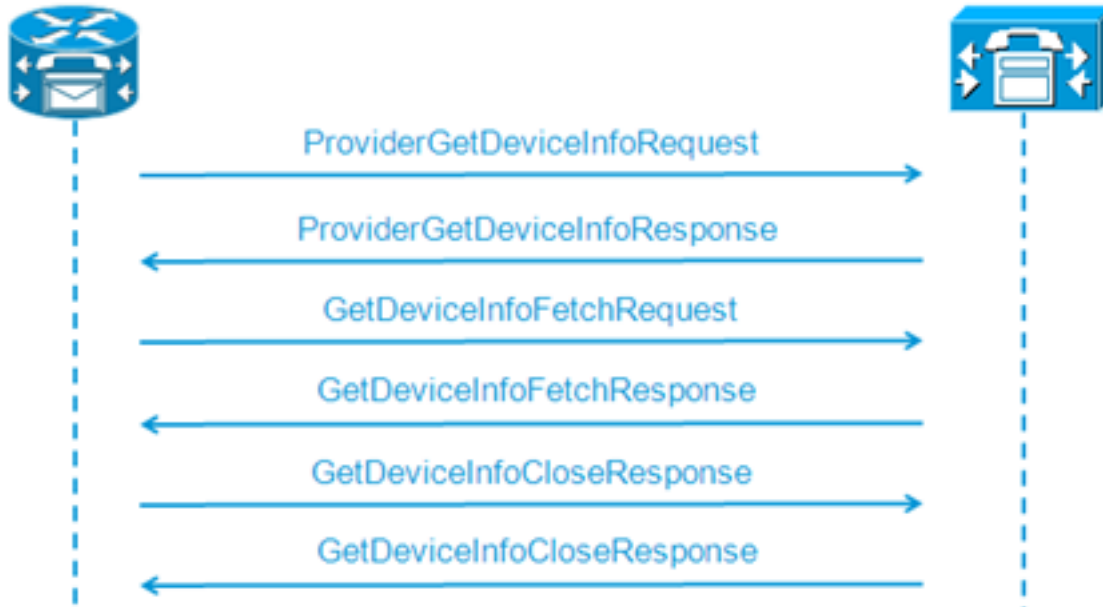
1. 打开供应商连接



```

21: 12:05:23.686 CST %JTAPI-CTIIMPL-7-UNK.(P1-tacjtapiuser) ProviderID =
P1-tacjtapiuser
22: 12:05:23.739 CST %JTAPI-CTIIMPL-7-UNK.(P1-tacjtapiuser) Trying to
create normal socket connection to 192.168.100.10
23: 12:05:23.747 CST %JTAPI-CTIIMPL-7-UNK.(P1-tacjtapiuser) connected
26: 12:05:24.112 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.ProviderOpenRequest {
provider = 192.168.100.10
qbeClientVersion = Cisco JTAPI 7.0(1.1000)-1 Release
login = com.cisco.cti.protocol.UnicodeString {
unicodedisplayName = tacjtapiuser
}
applicationID = Cisco IP IVR
desiredServerHeartbeatTime = 30
pluginName = CiscoJTAPI
}
28: 12:05:24.131 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Response: com.cisco.cti.protocol.ProviderOpenResponse {
sequenceNumber = 0
result = 0
providerInfoString = 7.1.5.10000-12
clientHeartbeat = 30
serverHeartbeat = 30
pluginVersion = 7.1.5.10000-2
pluginLocation = http://192.168.100.10/plugins/
providerId = 16777236
}
35: 12:05:24.858 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Event: com.cisco.cti.protocol.ProviderOpenCompletedEvent {
eventSequence = 0
reason = 0
providerInfoString = 7.1.5.10000-12
clientHeartbeat = 30
serverHeartbeat = 30
failureDescription = null
providerId = 16777236
}
  
```

2. 可控制的设备的查询



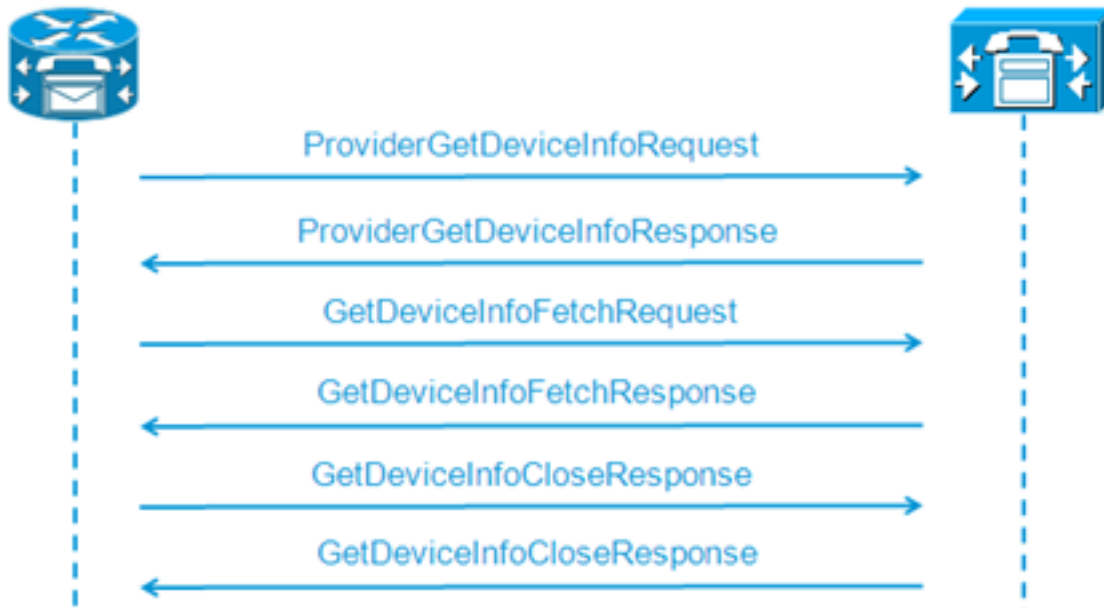
```

48: 12:05:24.864 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.ProviderGetDeviceInfoRequest {
sequenceNumber = 2
deviceGroup = 1
}
49: 12:05:24.865 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) received
Response: com.cisco.cti.protocol.ProviderGetDeviceInfoResponse {
sequenceNumber = 2
result = 0
}
50: 12:05:24.865 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.GetDeviceInfoFetchRequest {
sequenceNumber = 3
}
51: 12:05:25.011 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) received
Response: com.cisco.cti.protocol.GetDeviceInfoFetchResponse {
sequenceNumber = 3
result = 0
info = 2@[
com.cisco.cti.protocol.DeviceInfo {
name           = cue_ctiport1
type = 72
allowsRegistration = true
deviceID = 62
devTypeName = CTI Port
},
com.cisco.cti.protocol.DeviceInfo {
name           = cue_vm_ctirp
type = 73
allowsRegistration = true
deviceID = 61
devTypeName = CTI Route Point
}]
}
52: 12:05:25.012 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.GetDeviceInfoCloseRequest {
sequenceNumber = 4
}
53: 12:05:25.013 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10)
received Response: com.cisco.cti.protocol.GetDeviceInfoCloseResponse {
sequenceNumber = 4
}
54: 12:05:25.013 CST %JTAPI-MISC-7-UNK. (P1-192.168.100.10)

```

creating controlled devices

3. 得到CTI端口线路信息



```
55: 12:05:25.024 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.DeviceGetLineInfoRequest {
sequenceNumber = 5
deviceName = cue_ctiport1
}
56: 12:05:25.026 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10)
received Response: com.cisco.cti.protocol.DeviceGetLineInfoResponse {
sequenceNumber = 5
result = 0
}
57: 12:05:25.026 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.GetLineInfoFetchRequest {
sequenceNumber = 6
}
58: 12:05:25.029 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10)
received Response: com.cisco.cti.protocol.GetLineInfoFetchResponse {
sequenceNumber = 6
result = 0
com.cisco.cti.protocol.LineInfo {
name = 8501
displayName =
maxNumberOfCalls = 4
lineInstance = 1
unicodeDisplayName = com.cisco.cti.protocol.UnicodeString {
}
partition = cue_pt
defaultIntercomTargetInfo = com.cisco.cti.protocol.LineIntercomSpeedDialInfo {
}]
59: 12:05:25.029 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.GetLineInfoCloseRequest {
sequenceNumber = 7
}
60: 12:05:25.031 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10)
received Response: com.cisco.cti.protocol.GetLineInfoCloseResponse {
sequenceNumber = 7
result = 0
}
61: 12:05:25.042 CST %JTAPI-CTI-7-UNK. (P1-tacjtapiuser)
DeviceMap: adding device "cue_ctiport1"
4. 得到CTI RP线路信息 62: 12:05:25.043 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10)
[SS_TEL_INIT]
```

```

sending: com.cisco.cti.protocol.DeviceGetLineInfoRequest {
sequenceNumber = 8
deviceName = cue_vm_ctirp
}
63: 12:05:25.044 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10)
received Response: com.cisco.cti.protocol.DeviceGetLineInfoResponse {
sequenceNumber = 8
result = 0
}
64: 12:05:25.045 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.GetLineInfoFetchRequest {
sequenceNumber = 9
}
65: 12:05:25.047 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10)
received Response: com.cisco.cti.protocol.GetLineInfoFetchResponse {
sequenceNumber = 9
result = 0
info = 1@[
com.cisco.cti.protocol.LineInfo {
  name = 8000
displayName =
permanentLineID = 52
partition = cue_pt
defaultIntercomTargetInfo = com.cisco.cti.protocol.LineIntercomSpeedDialInfo {
unicodeLabel = com.cisco.cti.protocol.UnicodeString {
}
}
}
66: 12:05:25.048 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.GetLineInfoCloseRequest {
sequenceNumber = 10
}
67: 12:05:25.058 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10)
received Response: com.cisco.cti.protocol.GetLineInfoCloseResponse {
sequenceNumber = 10
result = 0
}
68: 12:05:25.059 CST %JTAPI-CTI-7-UNK. (P1-tacjtapiuser)
DeviceMap: adding device "cue_vm_ctirp"
69: 12:05:25.059 CST %JTAPI-CTI-7-UNK. (P1-192.168.100.10)
refreshing device map: previous=0 current=2 created=2 removed=0

```

5. 提示运用已接收配置 76: 12:05:25.064 CST %JTAPI-MISC-7-UNK.Provider 192.168.100.10

```

open, beginning device initialization
77: 12:05:25.071 CST %JTAPI-JTAPI-7-UNK. (P1-tacjtapiuser) [SS_TEL_INIT]
(P1-tacjtapiuser) Request: addObserver
79: 12:05:25.073 CST %JTAPI-MISC-7-UNK.ObserverThread
(com.cisco.wf.subsystems.jtapi.SubsystemJTAPI$ProviderObserver@3d823d82):created
80:12:05:25.074 CST %JTAPI-JTAPI-7-UNK. (P1-tacjtapiuser) ProvOutOfServiceEv [#0]
Cause:100 CallCtlCause:0 CiscoFeatureReason:12
82: 12:05:25.085 CST %JTAPI-MISC-7-UNK.ObserverThread
(com.cisco.wf.subsystems.jtapi.SubsystemJTAPI$ProviderObserver@3d823d82):
queuing com.cisco.jtapi.JtapiProviderEventSet
83: 12:05:25.084 CST %JTAPI-MISC-7-UNK. (P1-192.168.100.10)
ProviderRetryThread starting up
85: 12:05:25.084 CST %JTAPI-MISC-7-UNK.ObserverThread
(com.cisco.wf.subsystems.jtapi.SubsystemJTAPI$ProviderObserver@3d823d82)
starting up...
90: 12:05:25.102 CST %JTAPI-JTAPIIMPL-7-UNK.Partition Support 8000 in
partitioncue_pt
91: 12:05:25.102 CST %JTAPI-JTAPIIMPL-7-UNK. (P1-tacjtapiuser) cue_vm_ctirp:
Address: 8000 in partitioncue_pt created
92: 12:05:25.102 CST %JTAPI-JTAPIIMPL-7-UNK.Partition Internal Address Added

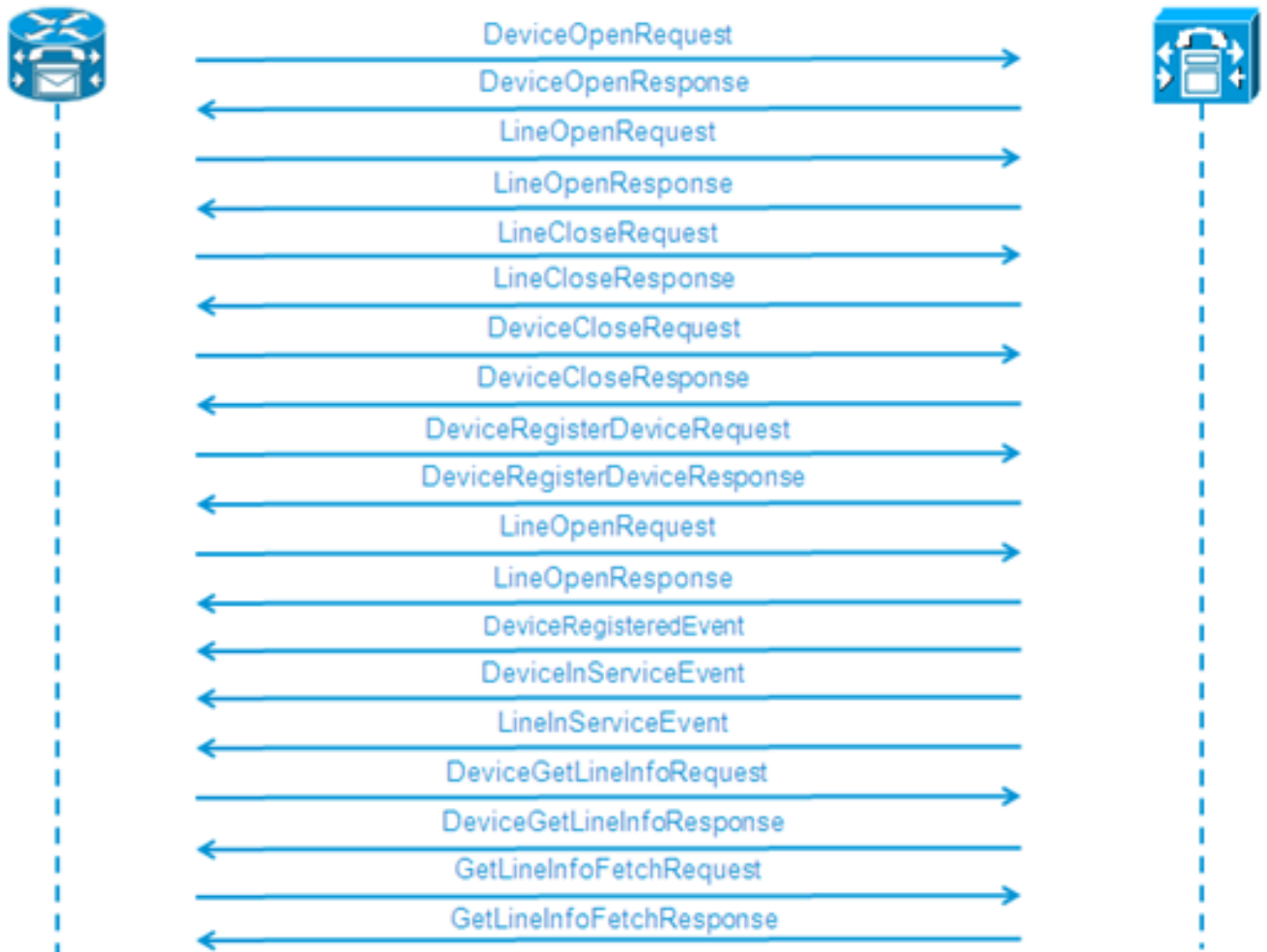
```

```

8000 in Partition cue_pt
93: 12:05:25.102 CST %JTAPI-JTAPIIMPL-7-UNK.Partition Support 8501 in
partitioncue_pt
94: 12:05:25.103 CST %JTAPI-JTAPIIMPL-7-UNK.(P1-tacjtapiuser) cue_ctiport1:
Address: 8501 in partitioncue_pt created
95: 12:05:25.103 CST %JTAPI-JTAPIIMPL-7-UNK.Partition Internal Address Added
8501 in Partition cue_pt
96: 12:05:25.103 CST %JTAPI-MISC-7-UNK.Provider "(P1-tacjtapiuser)" changing
state to IN_SERVICE
97: 12:05:25.103 CST %JTAPI-JTAPI-7-UNK.(P1-tacjtapiuser)[Thread-76]
(P1-tacjtapiuser) Request: getObservers
98: 12:05:25.103 CST %JTAPI-JTAPI-7-UNK.(P1-tacjtapiuser) ProvInServiceEv [#1]
Cause:100 CallCtlCause:0 CiscoFeatureReason:12
100: 12:05:25.103 CST %JTAPI-MISC-7-UNK.ObserverThread
(com.cisco.wf.subsystems.jtapi.SubsystemJTAPI$ProviderObserver@3d823d82):
queuing com.cisco.jtapi.JtapiProviderEventSet
101: 12:05:25.103 CST %JTAPI-JTAPIIMPL-7-UNK.Provider 192.168.100.10
initialized 2 devices
104: 12:05:25.104 CST %JTAPI-JTAPIIMPL-7-UNK:
[com.cisco.wf.subsystems.jtapi.SubsystemJTAPI$ProviderObserver@3d823d82]
delivering to providerChangedEvent
106: 12:05:25.523 CST %JTAPI-JTAPI-7-UNK.(P1-tacjtapiuser)[SS_TEL_INIT]
(P1-tacjtapiuser) Request: getAddress( 8501 )Partition = cue_pt
107: 12:05:25.526 CST %JTAPI-JTAPI-7-UNK.(P1-tacjtapiuser)[SS_TEL_INIT]
[cue_ctiport1]Request: addObserver
(com.cisco.wf.subsystems.jtapi.TAPIPortGroup$Port$AddressCallObserver@5d085d08)

```

6. 得到CTI设备和线路的控制



```

109: 12:05:25.528 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) [SS_TEL_INIT]
sending:
com.cisco.cti.protocol.DeviceOpenRequest {

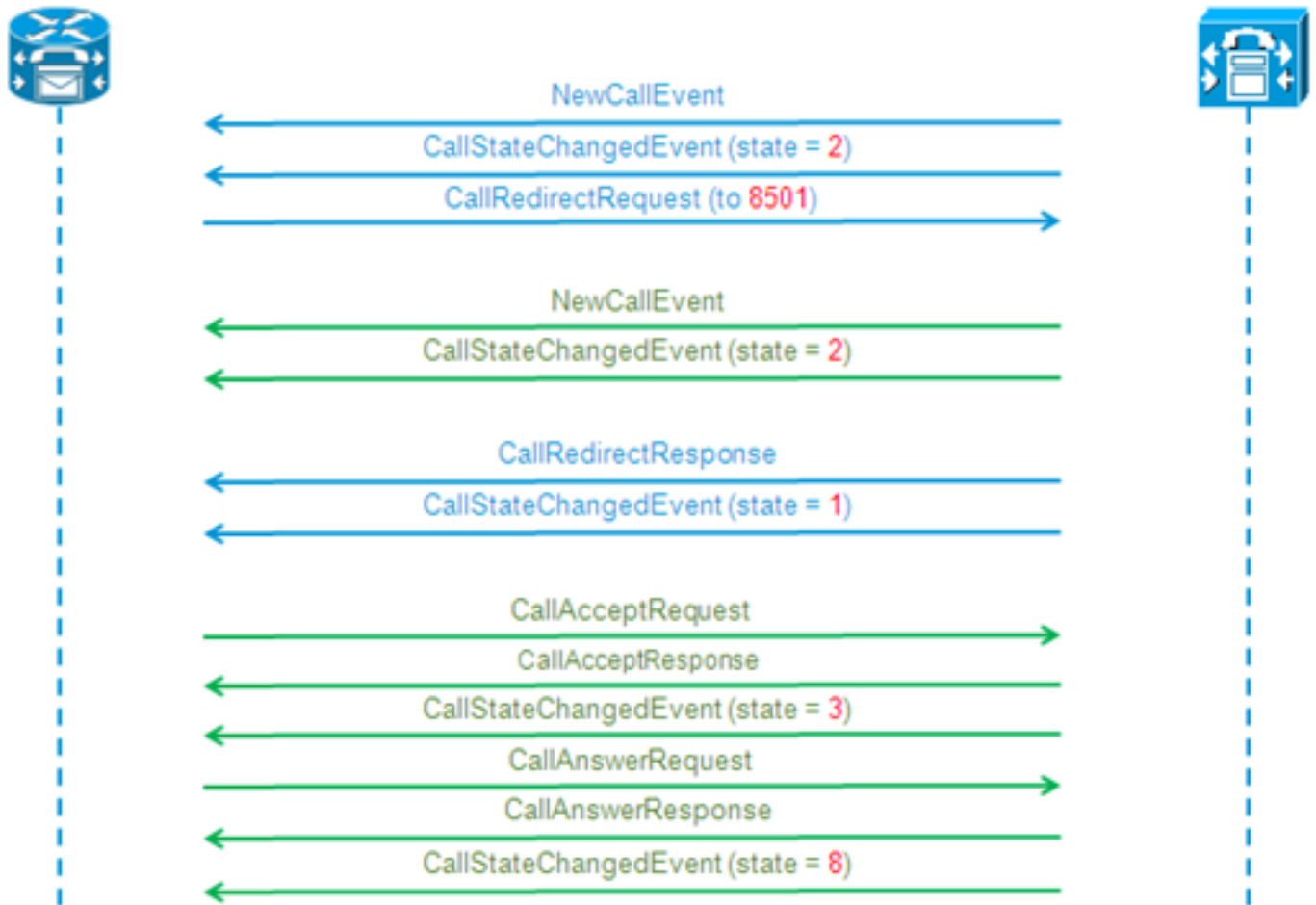
```

```
deviceName = cue_ctiport1
}
110: 12:05:25.533 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10)
received Response:
com.cisco.cti.protocol.DeviceOpenResponse {
result = 0
}
111: 12:05:25.533 CST %JTAPI-CTI-7-UNK.(P1-tacjtapiuser) DeviceMap: opening
device "cue_ctiport1"
112: 12:05:25.533 CST %JTAPI-JTAPIIMPL-7-UNK.(P1-tacjtapiuser) Terminal
"cue_ctiport1" out of service
113: 12:05:25.534 CST %JTAPI-JTAPI-7-UNK.(P1-tacjtapiuser) [cue_ctiport1]
CiscoTermOutOfServiceEv [#2] Cause:100 CallCtlCause:0 CiscoFeatureReason:12
119: 12:05:25.544 CST %JTAPI-JTAPIIMPL-7-UNK:Address [cue_ctiport1:8501:
cue_pt.(0,0)] out of service
120: 12:05:25.544 CST %JTAPI-JTAPI-7-UNK.(P1-tacjtapiuser) [8501:cue_pt]
CiscoAddrOutOfServiceEv [#3] Cause:100 CallCtlCause:0 CiscoFeatureReason:12
121: 12:05:25.546 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.LineOpenRequest {
deviceName      = cue_ctiport1
lineName       = 8501
}
122: 12:05:25.582 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Response: com.cisco.cti.protocol.LineOpenResponse {
134: 12:05:25.670 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.LineCloseRequest {
135: 12:05:25.673 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Response: com.cisco.cti.protocol.LineCloseResponse {
138: 12:05:25.674 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.DeviceCloseRequest {
139: 12:05:25.681 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Response: com.cisco.cti.protocol.DeviceCloseResponse {
141: 12:05:25.683 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.DeviceRegisterDeviceRequest {
deviceName = cue_ctiport1
142: 12:05:25.687 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Response: com.cisco.cti.protocol.DeviceRegisterDeviceResponse {
result = 0
name          = cue_ctiport1
allowsRegistration = true
}
143: 12:05:25.687 CST %JTAPI-CTI-7-UNK.(P1-tacjtapiuser) DeviceMap: opening
device "cue_ctiport1"
150: 12:05:25.688 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.LineOpenRequest {
deviceName = cue_ctiport1
lineName = 8501
151: 12:05:25.690 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Response: com.cisco.cti.protocol.LineOpenResponse {
152: 12:05:25.691 CST %JTAPI-JTAPIIMPL-7-UNK:cue_ctiport1: Lines opened
153: 12:05:25.739 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Event: com.cisco.cti.protocol.DeviceRegisteredEvent {
deviceInfo = com.cisco.cti.protocol.DeviceInfo {
allowsRegistration = true
controllable = true
}
156: 12:05:25.739 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) Received
DeviceRegisteredEvent
160: 12:05:25.740 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Event: com.cisco.cti.protocol.DeviceInServiceEvent {
162: 12:05:25.741 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Event: com.cisco.cti.protocol.LineInServiceEvent {
}
```

基本呼叫转发对语音邮件

(从CiscoJtapi1 /Cisco Jtapi2日志的输出)

新的呼叫和重定向对可用端口



新的呼叫和重定向对可用端口

```
12:46:00.396 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received Event:
com.cisco.cti.protocol.NewCallEvent {
deviceName           = cue_vm_ctirp
callLegID = 25626132
callID = 9040
callingParty = 3001
calledParty = 8000
callingPartyName = Ext 3001 - Phone
callingPartyDeviceName = SEP0023331C29EC
unModifiedCalledParty = 8000
unModifiedOriginalCalledParty = 8000
unModifiedLastRedirectingParty =
}
12:46:00.400 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received Event:
com.cisco.cti.protocol.CallStateChangedEvent {
callLegID = 25626132
state = 2
reason = 1
}
```

```
12:46:00.402 CST %JTAPI-CTI-7-UNK.(P1-tacjtapiuser){Line:cue_vm_ctirp:8000:
cue_pt.(1,28)
|Call:[GCID=
(9040/1),CID=25626132]} NewCall [ state=OFFERING auxData=1 destCM=1 destType=
IN_CLUSTER unModifiedCg=3001
unModifiedCd=8000 unModifiedOriginalCd=8000 unModifiedLastRedirected= calling=3001
callingName=Ext 3001 -
Phone called=8000 calledName= origParty=8000 origName= lastRedirected=
lastRedirectedName= origin=INBOUNDINTERNAL reason=DIRECTCALL activeTone=0
deviceName=cue_vm_ctirp bRemoteInUse=false bPrivacy=false CallSelectStatus=0
CallingPartyPI=True CallingPartyDisplayNamePI=True CalledPartyPI=True
CalledPartyDisplayNamePI=True OriginalCalledPartyPI=True]
12:46:00.424 CST %JTAPI-JTAPIIMPL-7-UNK:{(P1-tacjtapiuser) GCID=(1,9040)->ACTIVE}
Initializing to OFFERING for 8000:cue_pt Cause=CAUSE_NORMAL Reason= 1
12:46:00.424 CST %JTAPI-JTAPI-7-UNK:[[3001:cue_pt/(P1-tacjtapiuser) GCID=
(1,9040)->ACTIVE]->IDLE]creating external connection for 3001:cue_pt
12:46:00.424 CST %JTAPI-JTAPI-7-UNK:{ CcnCall=Call:[GCID=(9040/1),CID=25626132]
Connection=[3001:cue_pt/(P1-tacjtapiuser) GCID=(1,9040)->ACTIVE]->IDLE: creating
new Connection for CCNCall }
12:46:00.425 CST %JTAPI-JTAPI-7-UNK:[9040/1]CallImpl.deliverEvents(): for all
1 observers
12:46:00.430 CST %JTAPI-JTAPI-7-UNK.(P1-tacjtapiuser)[SS_TEL_ROUTE_CALL_EV][[
8000:cue_pt/(P1-tacjtapiuser) GCID=(1,9040)->ACTIVE]->OFFERED]Request: redirect
(8501, REDIRECT_NORMAL, DEFAULT_SEARCH_SPACE, CALLED_ADDRESS_UNCHANGED,
REDIRECT,8501,null,REDIRECT_WITHOUT_MODIFIED_CALLING_PARTY,1)
12:46:00.430 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10)
[SS_TEL_ROUTE_CALL_EV] sending: com.cisco.cti.protocol.CallRedirectRequest {
callLegID = 25626132
redirectAddress = 8501
unconditional = false
redirectReason = 0
preferredOriginalCalledParty = 8501
}
```

对CTI端口的新的呼叫

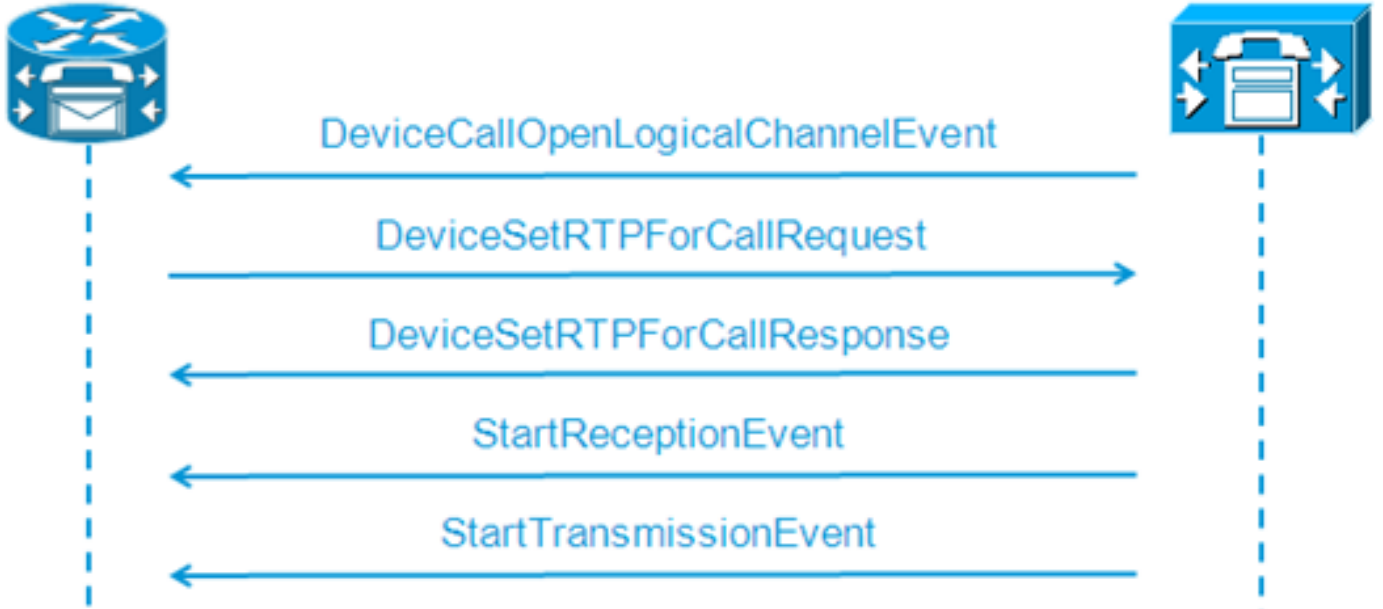
```
12:46:00.460 %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Event: com.cisco.cti.protocol.NewCallEvent {
deviceName = cue_ctiport1
callLegID = 25626133
callID = 9040
callingParty = 3001
calledParty = 8501
originalCalledParty = 8000
reason = 6
lastRedirectingParty = 8000
callingPartyDeviceName = SEP0023331C29EC
}
12:46:00.463 %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Event: com.cisco.cti.protocol.CallStateChangedEvent {
callLegID = 25626133
state = 2
}
12:46:00.464 %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Response: com.cisco.cti.protocol.CallRedirectResponse {
result = 0
}
12:46:00.468 %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Event: com.cisco.cti.protocol.CallStateChangedEvent {
callLegID = 25626132
state = 1
farEndpointSpecified = true
```

```
fwdDestinationAddress =
reason = 68501
callingParty = 3001
callingPartyName = Ext 3001 - Phone
calledParty = 8000 }
12:46:00.481 %JTAPI-JTAPIIMPL-7-UNK: {(P1-tacjtapiuser) GCID=(1,9040)->ACTIVE}
Initializing to OFFERING for 8501:cue_pt Cause=CAUSE_REDIRECTED Reason= 6
12:46:00.481 %JTAPI-JTAPIIMPL-7-UNK: {(P1-tacjtapiuser) GCID=(1,9040)->ACTIVE}
Received a redirected call -- lastRedAddress is 8000
12:46:00.487 %JTAPI-CTI-7-UNK. (P1-tacjtapiuser) {Line:cue_ctiport1:8501:cue_pt.
(1,24) |Call:[GCID=(9040/1),CID=25626133]} CallStateChanged [ state=OFFERING
cause=NOERROR]
12:46:00.489 %JTAPI-CTI-7-UNK. (P1-tacjtapiuser) {Line:cue_vm_ctirp:8000:cue_pt.
(1,28) |Call:[GCID=(9040/1),CID=25626132]} CallStateChanged [ state=IDLE cause=
NOERROR destType=IN_CLUSTER destCM=1 fwdDestination=8501]
```

CTI端口接受重定向呼叫

```
12:46:00.490 %JTAPI-JTAPI-7-UNK. (P1-tacjtapiuser) [SS_TEL_CALL_CONN_OFFERED:8501]
[[8501:cue_pt/(P1-tacjtapiuser) GCID=(1,9040)->ACTIVE]->OFFERED]Request: accept()
12:46:00.491 %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) [SS_TEL_CALL_
CONN_OFFERED:8501] sending: com.cisco.cti.protocol.CallAcceptRequest {
callLegID
                  = 25626133
}
12:46:00.495 %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) received Response:
com.cisco.cti.protocol.CallAcceptResponse {
result
          = 0
}
12:46:00.498 %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) received Event:
com.cisco.cti.protocol.CallStateChangedEvent {
callLegID
                  = 25626133
state = 3
}
12:46:00.499 %JTAPI-CTI-7-UNK. (P1-tacjtapiuser) {Line:cue_ctiport1:8501:cue_pt.
(1,24) |Call:[GCID=(9040/1),CID=25626133]} CallStateChanged [ state=ACCEPTED
cause=NOERROR]
12:46:00.502 %JTAPI-JTAPIIMPL-7-UNK. (P1-tacjtapiuser) Terminal "cue_ctiport1"
in service
12:46:00.503 %JTAPI-JTAPIIMPL-7-UNK: {(P1-tacjtapiuser) GCID=(1,9040)->ACTIVE}
Handling
External STATE_RINGBACK for 3001:cue_pt
12:46:00.517 %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10)
[ENG_TASK:0x98bca5a08_voicebrowser.aef] sending:
com.cisco.cti.protocol.CallAnswerRequest {
callLegID = 25626133
}
12:46:00.522 %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) received Response:
com.cisco.cti.protocol.CallAnswerResponse {
result
          = 0
}
12:46:00.530 %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) received Event:
com.cisco.cti.protocol.CallStateChangedEvent {
callLegID
                  = 25626133
state = 8
}
```

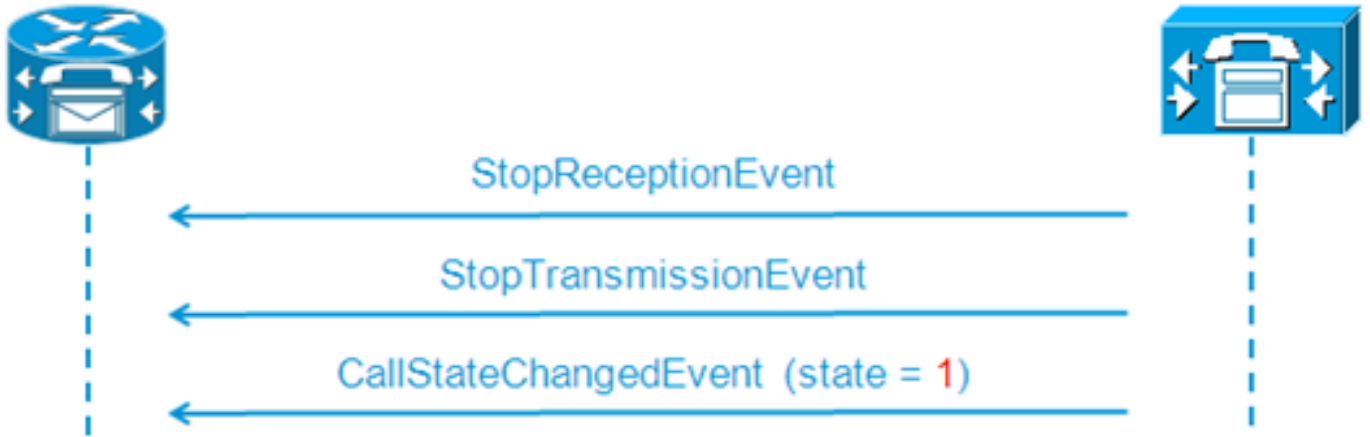
媒体协商



```

12:46:00.531 %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) received Event:
com.cisco.cti.protocol.DeviceCallOpenLogicalChannelEvent {
callLegID = 25626133
compressionType = 4
}
12:46:00.531 %JTAPI-CTI-7-UNK. (P1-tacjtapiuser) {Line:cue_ctiport1:8501:
cue_pt. (1,24) |Call: [GCID=(9040/1),CID=25626133]} CallStateChanged
[ state=CONNECTED cause=NOERROR]
12:46:00.537 %JTAPI-JTAPI-7-UNK. (P1-tacjtapiuser) [SS_TEL_OPEN_LOGICAL_CHANNEL:
8501][cue_ctiport1]
Request: setRTPParams(CiscoRTPParams192.168.105.224/16384)
12:46:00.537 %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) [SS_TEL_OPEN_
LOGICAL_CHANNEL:8501] sending:
com.cisco.cti.protocol.DeviceSetRTPForCallRequest {
callLegID = 25626133
ipAddress = -529946432
rtpPortNumber = 16384
}
12:46:00.540 %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) received Response:
com.cisco.cti.protocol.DeviceSetRTPForCallResponse {
result = 0
}
12:46:00.591 %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) received Event:
com.cisco.cti.protocol.StartReceptionEvent {
callLegID = 25626133
ipAddr = -529946432
rtpPortNumber = 16384
compressionType = 4
}
12:46:00.596 %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) received Event:
com.cisco.cti.protocol.StartTransmissionEvent {
callLegID = 25626133
ipAddr = -1167415104
rtpPortNumber = 22668
compressionType = 4
}
  
```

呼叫断开



```

12:46:09.438 %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) received Event:
com.cisco.cti.protocol.StopReceptionEvent {
callLegID = 25626133
}
12:46:09.438 %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) received Event:
com.cisco.cti.protocol.StopTransmissionEvent {
callLegID = 25626133
}
12:46:09.441 %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) received Event:
com.cisco.cti.protocol.CallStateChangedEvent {
callLegID                = 25626133
state = 1
cause = 16
}
12:46:09.443 %JTAPI-CTI-7-UNK. (P1-tacjtapiuser) {Line:cue_ctiport1:8501:
cue_pt. (1,24) |Call: [GCID=(9040/1),CID=25626133]} CallStateChanged
[ state=IDLE cause=NORMALCALLCLEARING]

```

MWI开/关信令

提示开MWI闪亮指示线路的3001

```

12:46:02.714 CST %JTAPI-JTAPI-7-UNK. (P1-tacjtapiuser) [Thread-88] [8501:cue_pt]
Request:
setMessageWaiting ( 3001,true )
12:46:02.714 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) [Thread-88]
sending: com.cisco.cti.protocol.LineSetMessageWaitingRequest {
sequenceNumber = 57
lineName       = 3001
lampMode = 2
}
12:46:02.718 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) received
Response: com.cisco.cti.protocol.LineSetMessageWaitingResponse {
sequenceNumber = 57
result = 0
}

```

删除从邮箱的消息的已拨号DTMF编号'3'

```

12:55:52.145 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) received Event:
com.cisco.cti.protocol.DtmfEvent {
eventSequence = 70
callLegID = 25626160
digit      = 3
}

```

```
12:55:52.145 CST %JTAPI-CTIIMPL-7-UNK. (P1-192.168.100.10) EventThread handling
event com.cisco.cti.protocol.DtmfEvent[70]
12:55:52.146 CST %JTAPI-CTI-7-UNK. () {Line:cue_ctiport1:8501:cue_pt.(1,64)|Call:
[GCID=(9047/1),CID=25626160]}
DTMF [digit=3]
```

提示为线路3001关闭MWI闪亮指示

```
12:55:52.209 CST %JTAPI-JTAPI-7-UNK. (P1-tacjtapiuser) [Thread-86] [8501:cue_pt]
Request: setMessageWaiting ( 3001,false )
12:55:52.209 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) [Thread-86] sending:
com.cisco.cti.protocol.LineSetMessageWaitingRequest {
sequenceNumber = 62
lineName           = 3001
lampMode = 1
}
12:55:52.212 CST %JTAPI-PROTOCOL-7-UNK. (P1-192.168.100.10) received Response:
com.cisco.cti.protocol.LineSetMessageWaitingResponse {
sequenceNumber = 62
result = 0
}
```

实时CCN日志

(从实时CCN日志的输出)

这是从此中前一个示例的同一呼叫如何出现，当实时CCN日志收集时。

呼叫建立

```
12:46:00.425 ACCN TELS 0 assigned STANDARD-worker-8
12:46:00.425 ACCN TELS 0 Route Connection=[8000:cue_pt/(P1-tacjtapiuser) GCID=
(1,9040)->ACTIVE]->OFFERED, reason=1...
12:46:00.426 ACCN TELS 0 Call.received() JTAPICallContact[id=7,type=Cisco JTAPI
Call,implId=9040/1,active=true,state=CALL_RECEIVED,inbound=true...
12:46:00.429 ACCN TELS 0 Route Connection: [8000:cue_pt/(P1-tacjtapiuser)
GCID=(1,9040)->ACTIVE]->OFFERED, CTI Port selected: TP[id=0,implId=8501,
state=IN_USE]
12:46:00.429 ACCN TELS 0 RouteCallObserver.callChangedEvent: redirecting to
8501, css=default
12:46:00.480 ACCN TELS 0 Call.associated() JTAPICallContact[id=7,type=Cisco
JTAPI Call,implId=9040/1,active=true,state=CALL_RECEIVED,
12:46:00.480 ACCN TELS 0 Route Connection: [8000:cue_pt/(P1-tacjtapiuser)
GCID=(1,9040)->ACTIVE]->OFFERED has 1 current sessions active.
12:46:00.484 ACCN TELS 0 CallID: 7, MediaID: 9040/1 CallCtlConnOfferedEv
received for CTI Port: 8501, lastRedirectedAddress: 8000
12:46:00.490 ACCN TELS 0 assigned STANDARD-worker-9
12:46:00.490 ACCN TELS 0 Route TR[num=8000], event=(P1-tacjtapiuser) 9040/1
CallCtlConnDisconnectedEv 8000:cue_pt [#108] Cause:100 CallCtlCause:0
CiscoCause:0 CiscoFeatureReason:6, cause=CAUSE_NORMAL[100],
meta=META_CALL_REMOVING_PARTY[131]
12:46:00.499 ACCN TELS 0 CallID: 7, MediaID: 9040/1 Accepting call for CTI
Route Point: 8000 on CTI Port: 8501, ciscoCause=31
12:46:00.501 ACCN TELS 0 Call.accepted() JTAPICallContact[id=7,type=Cisco
JTAPI Call,implId=9040/1,active=true,state=CALL_RECEIVED...
12:46:00.501 ACCN TELS 0 CallID:7 MediaId:9040/1, TerminalConnection to
Terminal: cue_ctiport1 is RINGING, [8501:cue_pt/(P1-tacjtapiuser)
GCID=(1,9040)->ACTIVE]->ALERTING
12:46:00.504 ACCN TELS 0 CallID:7 MediaId:9040/1 com.cisco.jtapi.
```

```
CiscoTermInServiceEvImpl received
12:46:00.504 ACCN TELS 0 TR[num=8000] Get TriggerMap[] return:
{secondaryDialogGroup=0, primaryDialogGroup=0}
12:46:00.513 ACCN TELS 0 Call.attributed() JTAPICallContact[id=7,type=Cisco
JTAPI Call,implId=9040/1,active=true,state=CALL_RECEIVED,...
12:46:00.513 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:4100000008 associated
with Task ID: 4100000008
12:46:00.533 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:4100000008,
TerminalConnection to Terminal:cue_ctiport1 is ACTIVE
12:46:00.534 ACCN TELS 0 Call.answered() JTAPICallContact[id=7,type=
Cisco JTAPI Call,implId=9040/1,active=true,state=CALL_ANSWERED,...
12:46:00.536 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:4100000008
com.cisco.jtapi.CiscoMediaOpenLogicalChannelEvImpl received
12:46:00.593 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:4100000008
com.cisco.jtapi.CiscoRTPInputStartedEvImpl received
12:46:00.597 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:4100000008
com.cisco.jtapi.CiscoRTPOutputStartedEvImpl received
```

呼叫断开

```
12:46:09.442 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:4100000008
com.cisco.jtapi.CiscoRTPInputStoppedEvImpl received
12:46:09.443 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:4100000008
com.cisco.jtapi.CiscoRTPOutputStoppedEvImpl received
12:46:09.447 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:4100000008
gets TermConnDroppedEv, meta code:132, cause code:100
12:46:09.447 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:4100000008,
TerminalConnection to Terminal: cue_ctiport1 is DROPPED, 9040/1
12:46:09.448 ACCN TELS 0 CallID:7 MediaId:9040/1 is removed from call session
mapping in Session[id=0x60db88402,parent=null,active=true,state=SESSION_IN_USE,
time=1354733160426], result:true
12:46:09.466 ACCN TELS 0 Call.abandoned() JTAPICallContact[id=7,type=Cisco
JTAPI Call,implId=9040/1,active=false,state=CALL_DISCONNECTED,...
12:46:09.466 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:4100000008, released TP
[type=Cisco CTI Port,id=0,implId=8501,active=false,state=IDLE] from 8000, and
releasing udpPort 16384
12:46:09.467 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:4100000008
com.cisco.jtapi.TermObservationEndedEvImpl received
```

排除故障案例研究

连接问题

在此方案中，提示端口和触发不向CUCM登记由于缺乏提示和CUCM之间的连接。

```
CUE# show log name CiscoJtapi1.log tail
!! or show log name CiscoJtapi2.log tail
456: 13:20:28.331 CDT %JTAPI-MISC-7-UNK.(P20-) started preloading classes
457: 13:20:28.331 CDT %JTAPI-MISC-7-UNK.(P20-) finished preloading classes
461: 13:20:28.331 CDT %JTAPI-CTI-7-UNK.(P20-) EventThread queue size
threshold is 25
462: 13:20:28.331 CDT %JTAPI-CTI-7-UNK.(P20-) Provider retry interval is set
to 30 seconds
463: 13:20:28.331 CDT %JTAPI-CTI-7-UNK.(P20-) Client desired server heartbeat
time is set to 30 seconds
464: 13:20:28.331 CDT %JTAPI-CTI-7-UNK.(P20-) CTI request timeout is is set to
30 seconds
465: 13:20:28.331 CDT %JTAPI-CTI-7-UNK.(P20-) Provider open request timeout
```

```

is set to 200 seconds
467: 13:20:28.331 CDT %JTAPI-CTI-7-UNK.(P20-) Provider Reconnect attempts is
set to 0
468: 13:20:28.331 CDT %JTAPI-CTI-7-UNK.(P20-) JAVA Socket Connect Timeout is
set to 15 seconds
469: 13:20:28.332 CDT %JTAPI-CTIIMPL-7-UNK.(P20-) Provider.info(CCMEncryption:
:encryptPassword was successful)
471: 13:20:28.334 CDT %JTAPI-JTAPIIMPL-7-UNK.ProviderImpl(): calling
jtapiProperties.getSecurityPropertyForInstance()
472: 13:20:28.334 CDT %JTAPI-JTAPIIMPL-7-UNK.(P20-tacjtapiuser ) TraceModule:
JTAPI version Cisco Jtapi version 7.0(1.1000)-1 Release
473: 13:20:28.334 CDT %JTAPI-JTAPIIMPL-7-UNK.(P20-tacjtapiuser ) Route Select
Timeout is 5000 msec
474: 13:20:28.335 CDT %JTAPI-JTAPIIMPL-7-UNK.(P20-tacjtapiuser ) Jtapi post
condition timeout is set to 15 seconds
476: 13:20:28.335 CDT %JTAPI-CTIIMPL-7-UNK.(P20-tacjtapiuser ) Opening server
"192.168.100.10" login "tacjtapiuser "
477: 13:20:28.335 CDT %JTAPI-CTIIMPL-7-UNK.(P20-tacjtapiuser ) ProviderID =
P20-tacjtapiuser 478: 13:20:28.337 CDT %JTAPI-CTIIMPL-7-UNK.(P20-tacjtapiuser )
Trying to create normal socket connection to 192.168.100.10
479: 13:20:38.338 CDT %JTAPI-JTAPI-7-UNK:[DefaultJtapiPeer]PlatformExceptionImpl
caught:
Unable to create provider --

```

注意：时间戳秒钟去从13:20:28到13:20:38;因此，我们能告诉提示没有能在无法的确认前打开TCP socket 10秒创建供应商。

验证问题

在此方案中，因为密码配置在提示和CUCM之间不匹配，提示失效和触发向CUCM登记。

CCN日志

```

CUE# show trace buffer tail
Press CTRL-C to exit...
140053.173 ACCN TELS 0 TAPIPortGroup Leaving getActiveCCM(), retvalnull
140123.184 ACCN TELS 0 TAPIPortGroup Enter getActiveCCM()
140123.184 ACCN TELS 0 TAPIPortGroup getActiveCCM() subsystemstate3
140123.184 ACCN TELS 0 TAPIPortGroup getActiveCCM() subsystemJTAPI is not
in service or partial service
140123.184 ACCN TELS 0 TAPIPortGroup Leaving getActiveCCM(), retvalnull
atrace.log

14:12:18.681 ACCN TELS 0 JTAPI_PROVIDER_EVENT:JTAPI Provider state is changed:
JTAPI provider name=192.168.100.10,Event=ProvShutdownEv received
14:12:18.682 ACCN TELS 0 SS_LOGIN:JTAPI Login String: Module=JTAPI Subsystem,
JTAPI login string=192.168.100.10;login=tacjtapiuser ;passwd=****;appinfo=
Cisco IP IVR
14:12:18.682 ACCN TELS 0 PROVIDER_CLEANUP:Cleaning up JTAPI provider:
Module=JTAPI Subsystem,JTAPI provider name=192.168.100.10
14:12:18.682 ACCN TELS 0 TAPIPortGroup 1 getNumPorts() for Cisco CTI Port = 2
14:12:18.682 ACCN TELS 0 TPG[id=1,state=PARTIAL_SERVICE] removeRoute() -
TR[num=9500]
14:12:18.682 ACCN TELS 0 TPG[id=1,state=PARTIAL_SERVICE] removeRoute() -
TR[num=9000]
14:12:18.682 ACCN TELS 0 MwiAddress.clear: [addrStr=, addr=null, inService=false,
isRegistered=false]
14:12:18.682 ACCN TELS 0 MwiAddress.unregister: [addrStr=, addr=null,
inService=false, isRegistered=false]
14:12:18.682 ACCN TELS 0 TAPIPortGroup 1 getNumPorts() for Cisco CTI Port = 0
14:12:18.682 ACCN TELS 0 Number of CTI ports = 0

```

```
14:12:18.682 ACCN TELS 0 calculateSubsystemState
14:12:18.682 ACCN TELS 0 TPG[id=1,state=PARTIAL_SERVICE] Triggers: ISV = 0,
OOS = 0, PARTIAL = 0
14:12:18.682 ACCN TELS 0 TAPIPortGroup 1 getNumPorts() for Cisco CTI Port = 0
14:12:18.682 ACCN TELS 0 calculateSubsystemState -> Groups: ISV = 0, OOS = 0,
PARTIAL/OTHERS = 1
14:12:18.682 ACCN TELS 0 calculateSubsystemState -> Triggers: ENABLED = 0,
DISABLED = 2, CONFIG ERR = 0
14:12:18.682 ACCN TELS 0 calculateSubsystemState -> subsystem partial in
service, unchanged cause:
A number of route points are OOS - TR[num=9000], TR[num=9500]; A number of
CTI ports are OOS - TPG[id=1,state=PARTIAL_SERVICE].Ports[9590]
14:12:18.689 ACCN TELS 0 SS_PARTIAL_SERVICE:JTAPI subsystem in partial service:
Failure reason=A number of route points are OOS - TR[num=9000], TR[num=9500];
A number of CTI ports are OOS - TPG[id=1,state=PARTIAL_SERVICE].Ports[9590]
14:12:18.689 ACCN TELS 0 GET_NEW_PROVIDER:Attempt to get JTAPI provider
14:12:18.693 ACCN TELS 0 Calling updateJTAPIPackage: 192.168.100.10
Module=JTAPI_PROVIDER_INIT,Exception=com.cisco.jtapi.PlatformExceptionImpl:
Unable to create provider
-- bad login or password.
14:12:18.828 ACCN TELS 0 EXCEPTION:com.cisco.jtapi.PlatformExceptionImpl:
Unable to create provider
-- bad login or password.
```

CiscoJtapi1.log/CiscoJtapi2.log

```
6318: 14:22:26.653 CDT %JTAPI-CTIIMPL-7-UNK.(P62-tacjtapiuser ) Trying to
create normal socket connection to 192.168.100.10
6319: 14:22:26.654 CDT %JTAPI-CTIIMPL-7-UNK.(P62-tacjtapiuser ) connected
6321: 14:22:26.654 CDT %JTAPI-PROTOCOL-7-UNK.(P62-192.168.100.10)
[SS_TEL_REINIT] sending: com.cisco.cti.protocol.ProviderOpenRequest {
provider = 192.168.100.10
qbcClientVersion = Cisco JTAPI 7.0(1.1000)-1 Release
login = com.cisco.cti.protocol.UnicodeString {
unicodedisplayName = tacjtapiuser
}
filter = com.cisco.cti.protocol.ProviderEventFilter {
deviceRegistered = true
deviceUnregistered = true
desiredServerHeartbeatTime = 30
}
6331: 14:22:26.781 CDT %JTAPI-PROTOCOL-7-UNK(P62-192.168.100.10)
received Event: com.cisco.cti.protocol.ProviderOpenCompletedEvent {
eventSequence = 251
reason = -1932787616
providerInfoString = 7.1.2.21900-5
failureDescription = Directory login failed - authentication failed.
providerId = 16777255
}
6333: 14:22:26.781 CDT %JTAPI-PROTOCOL-7-UNK.(P62-192.168.100.10)
received Event: com.cisco.cti.protocol.ProviderClosedEvent {
eventSequence = 252
reason = 4
}
6338: 14:22:26.781 CDT %JTAPI-PROTOCOL-7-UNK.(P62-192.168.100.10)
Received ProviderClosedEvent
6339: 14:22:26.781 CDT %JTAPI-PROTOCOL-7-UNK.(P62-192.168.100.10)
received Event: com.cisco.cti.protocol.ProviderOutOfServiceEvent {
eventSequence = 253
PROVIDER_OUT_OF_SERVICE_EVENT = 200
}
6343: 14:22:26.782 CDT %JTAPI-JTAPI-7-UNK:[DefaultJtapiPeer]
PlatformExceptionImpl caught: Unable to create provider -- bad login or password.
6344: 14:22:26.881 CDT %JTAPI-CTIIMPL-7-UNK.(P62-192.168.100.10) ReceiveThread:
```

caught java.net.SocketException: **The socket was closed**

用户没CTI启用

在此方案中，因为JTAPI应用程序用户未被添加到CUCM侧的，英文虎报CTI启用的权限组提示失效和触发向CUCM登记。所以，既使当用户凭证相应地验证，JTAPI用户，在这种情况下 **tacjtapiuser**，不能通过CTI和JTAPI控制任何设备。

CiscoJtapi1.log/CiscoJtapi2.log

```
11590:14:41:08.768 CDT %JTAPI-PROTOCOL-7-UNK. (P115-192.168.100.10)
[ProviderRetryThread] sending:
com.cisco.cti.protocol.ProviderOpenRequest {
provider = 192.168.100.10
qbeClientVersion = Cisco JTAPI 7.0(1.1000)-1 Release
login = com.cisco.cti.protocol.UnicodeString {
unicodedisplayName = tacjtapiuser
}
applicationID = Cisco IP IVR
desiredServerHeartbeatTime = 30
requestTimer = 0
cmAssignedApplicationID = 0
pluginName = CiscoJTAPI
}
11593:14:41:08.770 CDT %JTAPI-PROTOCOL-7-UNK. (P115-192.168.100.10)
received Response: com.cisco.cti.protocol.ProviderOpenResponse {
sequenceNumber = 117
result = 0
providerInfoString = 7.1.2.21900-5
clientHeartbeat = 30
serverHeartbeat = 30
requestTimer = 5
pluginVersion = 7.1.2.10000-5
pluginLocation = http://192.168.100.10/plugins/
providerId = 16777220
}
11600: 14:41:08.899 CDT %JTAPI-PROTOCOL-7-UNK. (P115-192.168.100.10)
received Event: com.cisco.cti.protocol.ProviderOpenCompletedEvent {
eventSequence = 461
reason = -1932787617
sequenceNumber = 117
failureDescription = Directory login failed - User not present in Standard
CTI Users group.
providerId = 16777220
}
11608:14:41:08.900 CDT %JTAPI-PROTOCOL-7-UNK. (P115-192.168.100.10)
received Event:
com.cisco.cti.protocol.ProviderOutOfServiceEvent {
eventSequence = 463
PROVIDER_OUT_OF_SERVICE_EVENT = 200
}
```

CUCM CTI Manager服务发生故障

在此方案中，因为CUCM CTI Manager服务下降或在一种异常状态，提示端口和触发不能注册。它收到提示的连接尝试的一个“连接拒绝的”错误对JTAPI TCP端口2748。

```
18956: 16:25:45.516 CDT %JTAPI-CTIIMPL-7-UNK. (P200-) Provider.
info(CCMEncryption::encryptPassword was successful)
18957: 16:25:45.516 CDT %JTAPI-CTIIMPL-7-UNK. (P200-) application did
```

```

not set appinfo, creating default
18958: 16:25:45.516 CDT %JTAPI-JTAPIIMPL-7-UNK.ProviderImpl(): calling
jtapiProperties.getSecurityPropertyForInstance()
18959: 16:25:45.516 CDT %JTAPI-JTAPIIMPL-7-UNK.(P200-tacjtapiuser )
TraceModule: JTAPI version Cisco Jtapi version 7.0(1.1000)-1 Release
18960: 16:25:45.516 CDT %JTAPI-JTAPIIMPL-7-UNK.(P200-tacjtapiuser )
Route Select Timeout is 5000 msec
18961: 16:25:45.516 CDT %JTAPI-JTAPIIMPL-7-UNK.(P200-tacjtapiuser )
Jtapi post condition timeout is set
to 15 seconds
18962: 16:25:45.516 CDT %JTAPI-JTAPIIMPL-7-UNK.(P200-tacjtapiuser )
IgnoreFwdDestination
set to false
18963: 16:25:45.516 CDT %JTAPI-CTIIMPL-7-UNK.(P200-tacjtapiuser )
Opening server "192.168.100.10" login "tacjtapiuser "
18964: 16:25:45.516 CDT %JTAPI-CTIIMPL-7-UNK.(P200-tacjtapiuser )
ProviderID = P200-tacjtapiuser
18965: 16:25:45.517 CDT %JTAPI-CTIIMPL-7-UNK.(P200-tacjtapiuser )
Trying to create normal socket connection to 192.168.100.10
18966: 16:25:45.518 CDT %JTAPI-JTAPI-7-UNK:[DefaultJtapiPeer]
PlatformExceptionImpl caught:
Unable to create provider -- 192.168.100.10/192.168.100.10:2748 -
Connection refused

```

配置不匹配

在此方案中，提示不能注册JTAPI触发用第9999，因为应该匹配没有配置的CTI RP，或者未被添加到？可控制的设备？对于在CUCM侧的用户。提示认识到此，在接收从CUCM的**GetDeviceInfoFetchResponse**并且注意没有在运营商域后的一个设备，由该用户是指所有可控制的设备，将匹配触发编号配置本地。提示然后不设法发送该特定触发的一**DeviceOpenRequest**和只报告在跟踪的例外。提示仍然设法注册在CUCM发送的运营商域内的所有其它设备。

```

13:27:58.864 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received Response:
com.cisco.cti.protocol.GetDeviceInfoFetchResponse {
com.cisco.cti.protocol.DeviceInfo {
name = cue_vm_ctirp
}
}
13:27:58.960 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.DeviceGetLineInfoRequest {
deviceName = cue_vm_ctirp
}
13:27:58.962 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.GetLineInfoFetchRequest
13:27:58.964 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received Response:
com.cisco.cti.protocol.GetLineInfoFetchResponse{
name = 8000
}
13:27:58.966 CST %JTAPI-CTI-7-UNK(P1-tacjtapiuser) DeviceMap: adding device
"cue_vm_ctirp"
13:27:59.427 CST %JTAPI-JTAPI-7-UNK: InvalidArgumentExceptionImpl caught:
Address 9999 is not in provider's domain.

```

注意：即使当触发9999在提示配置本地，它不作为从CUCM接收的运营商域的部分，并且，不注册。

提示继续开放线路8000;哪些包括在供应商？s域

```

13:28:00.953 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.DeviceOpenRequest {
deviceName = cue_vm_ctirp
}

```

```

13:28:00.979 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.LineOpenRequest {
deviceName = cue_vm_ctirp
lineName = 8000
13:28:00.983 CST %JTAPI-JTAPIIMPL-7-UNK:cue_vm_ctirp: Lines opened
13:28:00.997 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.DeviceRegisterDeviceRequest
deviceName = cue_vm_ctirp
13:28:01.000 CST %JTAPI-CTI-7-UNK.(P1-tacjtapiuser) DeviceMap: opening device
"cue_vm_ctirp"
13:28:01.001 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.LineOpenRequest {
lineName = 8000
13:28:01.012 CST %JTAPI-JTAPIIMPL-7-UNK:cue_vm_ctirp: Lines opened
13:28:01.164 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received Event:
com.cisco.cti.protocol.DeviceRegisteredEvent {
13:28:01.165 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received Event:
com.cisco.cti.protocol.DeviceInServiceEvent {
13:28:01.166 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received Event:
com.cisco.cti.protocol.LineInServiceEvent {
13:28:01.168 CST %JTAPI-JTAPIIMPL-7-UNK.(P1-tacjtapiuser) Terminal
"cue_vm_ctirp" in service

```

CUCM呼叫路由问题

在此方案中，有DN的3001用户呼叫提示检查他的VM。呼叫被提交对提示的VM试验(CTI RP)与DN 8000。提示然后请求呼叫重新定向到其有DN的8501媒体CTI端口，但是呼叫不能重新定向，因为为DN配置的CSS 3001不访问CTI端口的DN分配的PT。

```

12:56:01.392 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Event: com.cisco.cti.protocol.NewCallEvent {
deviceName = cue_vm_ctirp
callLegID = 25626135
callID = 9041
callingParty = 3001
calledParty = 8000
originalCalledParty state = 2
}
12:56:01.404 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10)
[SS_TEL_ROUTE_CALL_EV] sending: com.cisco.cti.protocol.CallRedirectRequest {
callLegID = 25626135
redirectAddress = 8501
}
12:56:01.397 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Event: com.cisco.cti.protocol.CallStateChangedEvent {
callLegID = 25626135
state = 2
}
12:56:01.450 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Response: com.cisco.cti.protocol.FailureResponse {
result = -1932787660
description = redirect failure
}
12:56:01.450 CST %JTAPI-JTAPI-7-UNK: [[8000:cue_pt/(P1-tacjtapiuser)
GCID=(1,9041)->ACTIVE]->OFFERED]InvalidPartyExceptionImpl caught:
Request failed because of an invalid destination.
12:56:05.456 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Event: com.cisco.cti.protocol.CallStateChangedEvent {
callLegID = 25626135
state = 1
cause = 17
}

```



```

12:56:05.456 CST %JTAPI-CTI-7-UNK.(P1-tacjtapiuser){Line:cue_vm_ctirp:
8000:cue_pt.(1,28)|Call:[GCID=(9041/1),CID=25626135]}CallStateChanged
[ state=IDLE cause=USERBUSY]
12:56:05.457 CST %JTAPI-CTI-7-UNK:{ALL EXTERNAL ADDRESSES|Call(P1-tacjtapiuser)
GCID=(1,9041)->ACTIVE} ExternalCallStateChanged
[ state=IDLE cause=17 processEvent= reason =1 ]
12:56:05.457 CST %JTAPI-JTAPI-7-UNK.(P1-tacjtapiuser) 9041/1 ConnDisconnectedEv
3001:cue_pt [#160]
Cause:17 CallCtlCause:0 CiscoCause:17 CiscoFeatureReason:12
12:56:05.457 CST %JTAPI-JTAPI-7-UNK.(P1-tacjtapiuser)[(P1-192.168.100.10)
EventThread][SEP0023331C29EC] Request: getCallingTerminal()
12:56:05.457 CST %JTAPI-JTAPI-7-UNK.(P1-tacjtapiuser) 9041/1
CallCtlConnDisconnectedEv 3001:cue_pt [#161] Cause:17 CallCtlCause:0
CiscoCause:17 CiscoFeatureReason:12= 8000

```

许可证问题

在此方案中，因为VM端口的许可证未激活，提示无法注册其端口和触发。注册尝试在捕获没有被看到由于同一个原因。

从解码的atrace.log的摘要：

```

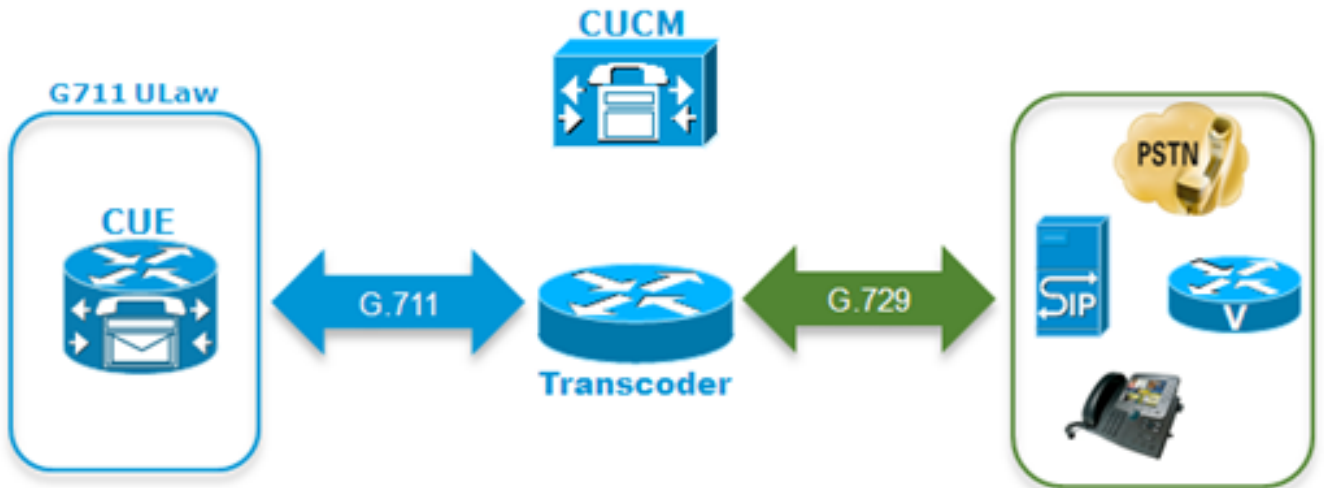
2551 11:45:17.178 LLMA LAPI 0 Llama: getMaxVmPortCount():
2547 11:45:17.178 LLMA LVMP 0 LlamaVmPortQuery: get(): maxCount
2551 11:45:17.178 LLMA LSDB 0 Llama: getMaxVmPortCount(): LlamaSysdbUser():
getInt(): Getting int /sw/apps/limitsManager/vmPort/query/maxCount returns 0
2551 11:45:17.178 LLMA LAPI 0 Llama: getMaxVmPortCount(): count: 0
2551 11:45:17.178 WFSP JTRG 0 WFSysdbNdJtapiTrg::getMaxSessions for trigger
for app: voicemail 0
2551 11:45:17.178 WFSP JTRG 0 WFSysdbNdJtapiTrg::commit warning session
value exceeded license max
2551 11:45:17.181 WFSP JTRG 0 com.cisco.aesop.sysdb.xactSysdbException:
Session value exceeds license limit
2551 11:45:19.654 LLMA LVMM 0 LlamaVmMbxQuery: get(): licenseStatus
2575 11:45:19.654 LLMA LSDB 0 Llama: showVoicemail(): LlamaSysdbUser():
getInt(): Getting int /sw/apps/limitsManager/vmMbx/query/licenseStatus returns 2
2575 11:45:19.657 LLMA LLMT 0 voicemail disabled, voicemail mailbox
activation count has been set to zero
3456 11:45:23.114 LLMA LAPI 0 Llama: getMaxPortCount():
2555 11:45:23.114 LLMA LPRT 0 LlamaPortQuery: get(): maxCount
3456 11:45:23.115 LLMA LSDB 0 Llama: getMaxPortCount(): LlamaSysdbUser():
getInt(): Getting int/sw/apps/limitsManager/port/query/maxCount returns 0
3456 11:45:23.115 LLMA LAPI 0 Llama: getMaxPortCount(): count: 0
3456 11:45:28.727 ACCN TELS 0 CueCiscoCall:getMajorVer() jtapi version=
7.0(1.1000)-1 majorVer=7
3456 11:45:28.785 ACCN TELS 0 JTAPI Login Str:
192.168.100.10;login=tacjtapiuser ;passwd=****;appinfo=Cisco IP IVR
3456 11:45:28.785 ACCN TELS 0 Actual Login Str:
192.168.100.10;login=tacjtapiuser ;passwd=cisco;appinfo=Cisco IP IVR
3477 11:45:31.330 ACCN TELS 0 Got JTAPI provider: Cisco Jtapi version
7.0(1.1000)-1 Release
3621 11:45:31.338 ACCN TELS 0 JTAPI_PROVIDER_EVENT:JTAPI Provider
state is changed: JTAPI provider name=192.168.100.10,Event=
ProvOutOfServiceEv received
3621 11:45:31.352 ACCN TELS 0 JTAPI_PROVIDER_EVENT:JTAPI Provider state
is changed: JTAPI provider name=192.168.100.10,Event=ProvInServiceEv received
3621 11:45:31.353 ACCN ATJT 0 checkConnectivity:
urlString:http://192.168.100.10/CCMpluginsServer/CiscoJTAPIClient.exe
3477 11:45:34.130 ACCN TELS 0 SS_OUT_OF_SERVICE:JTAPI subsystem in
out of service: Failure reason=A number of route points are OOS; A number of
CTI ports are OOS - all ports in TPG
3751 11:45:48.558 ACCN TELS 0 TAPIPortGroup: getActiveCCM() subsystemJTAPI
is not in service or partial service

```

最佳实践

提示只支持G711ulaw编码;因此在几乎每部署代码转换器要求为了提示能与使用不同的编码的其它设备或中继联络(包括G711alaw)。同样申请相互作用用只支持在波段之内DTMF媒介终接点(MTP)资源也要求的设备的DTMF。由于这些限制，思科推荐对：

- 创建仅提示使用情况的一隔离设备池？s CTI RP和CTI端口。在超过一个提示集成与CUCM情况下，然后请创建一设备池每个提示。
- 创建一个分开的仅区域提示的RP和端口的并且应用它给该隔离设备池。
- 保证区域配置允许与其他地区的仅G711。
- 保证一Media Resource Group List (MRGL)与联机应用给提示的CTI RP和CTI端口的设备池的转码资源，以便这些访问转码器资源，当需要。



- 如果用户无法通过与DTMF音的语音菜单导航，则很可能，必须添加MTP资源到提示设备的MRGL。

创建提示的一分开的VM配置文件在CUCM

为了避免一些最近的问题观察与CUCM CTI Manager，推荐关联所有电话对在CUCM侧的提示的JTAPI用户，而不是仅CTI RP和端口。

如果抗损远程站点电话(SRST)功能希望：

- 保证对应的SIP触发为在提示的每JTAPI触发配置。
- 保证被添加到分支路由器为了允许呼叫将路由的dial-peer到提示模块通过SIP，当在SRST模式时。
- 配置其中每一的外线号码掩码个CTI路由点，以及CFU (未注册的呼叫转发)字段的掩码保证CUCM的路由被注定的呼叫分支机构模块的到本地公共交换电话网(PSTN)网关，当CUCM和提示之间的连接丢失时或，如果自动化的备用路由(AAR)被调用。另外的翻译规则也许要求为了分支路由器能从PSTN路由呼入呼叫到提示模块。
- 如果对VM配置方法的直接交接是存在CUCM侧，并且用户在CME-SRST时要维护此功能，当，则使用CME的必须以转发所有呼叫(CFA)配置方法使用旧有假DN，在TransferToVM软键变得可用前。参考的[转移一个呼叫方直接地到Unity Express邮箱里](#)欲知更多信息。这是示例这如何能查找。如果使用CME-SRST而不是传统SRSTwith call-manager-fallback，请记住此能只执行。假设Dns在范围200-299。呼叫为x201进来。配置一ephone-dn用此命令：2551

```
11:45:17.178 LLMA LAPI 0 Llama: getMaxVmPortCount():
2547 11:45:17.178 LLMA LVMP 0 LlamaVmPortQuery: get(): maxCount
2551 11:45:17.178 LLMA LSDB 0 Llama: getMaxVmPortCount(): LlamaSysdbUser():
getInt(): Getting int /sw/apps/limitsManager/vmPort/query/maxCount returns 0
```

```

2551 11:45:17.178 LLMA LAPI 0 Llama: getMaxVmPortCount(): count: 0
2551 11:45:17.178 WFSP JTRG 0 WFSysdbNdJtapiTrg::getMaxSessions for trigger
for app: voicemail 0
2551 11:45:17.178 WFSP JTRG 0 WFSysdbNdJtapiTrg::commit warning session
value exceeded license max
2551 11:45:17.181 WFSP JTRG 0 com.cisco.aesop.sysdb.xactSysdbException:
Session value exceeds license limit
2551 11:45:19.654 LLMA LVMM 0 LlamaVmMbxQuery: get(): licenseStatus
2575 11:45:19.654 LLMA LSDB 0 Llama: showVoicemail(): LlamaSysdbUser():
getInt(): Getting int /sw/apps/limitsManager/vmMbx/query/licenseStatus returns 2
2575 11:45:19.657 LLMA LLMT 0 voicemail disabled, voicemail mailbox
activation count has been set to zero
3456 11:45:23.114 LLMA LAPI 0 Llama: getMaxPortCount():
2555 11:45:23.114 LLMA LPRT 0 LlamaPortQuery: get(): maxCount
3456 11:45:23.115 LLMA LSDB 0 Llama: getMaxPortCount(): LlamaSysdbUser():
getInt(): Getting int/sw/apps/limitsManager/port/query/maxCount returns 0
3456 11:45:23.115 LLMA LAPI 0 Llama: getMaxPortCount(): count: 0
3456 11:45:28.727 ACCN TELS 0 CueCiscoCall:getMajorVer() jtapi version=
7.0(1.1000)-1 majorVer=7
3456 11:45:28.785 ACCN TELS 0 JTAPI Login Str:
192.168.100.10;login=tacjtapiuser ;passwd=****;appinfo=Cisco IP IVR
3456 11:45:28.785 ACCN TELS 0 Actual Login Str:
192.168.100.10;login=tacjtapiuser ;passwd=cisco;appinfo=Cisco IP IVR
3477 11:45:31.330 ACCN TELS 0 Got JTAPI provider: Cisco Jtapi version
7.0(1.1000)-1 Release
3621 11:45:31.338 ACCN TELS 0 JTAPI_PROVIDER_EVENT:JTAPI Provider
state is changed: JTAPI provider name=192.168.100.10,Event=
ProvOutOfServiceEv received
3621 11:45:31.352 ACCN TELS 0 JTAPI_PROVIDER_EVENT:JTAPI Provider state
is changed: JTAPI provider name=192.168.100.10,Event=ProvInServiceEv received
3621 11:45:31.353 ACCN ATJT 0 checkConnectivity:
urlString:http://192.168.100.10/CCMPluginsServer/CiscoJTAPIClient.exe
3477 11:45:34.130 ACCN TELS 0 SS_OUT_OF_SERVICE:JTAPI subsystem in
out of service: Failure reason=A number of route points are OOS; A number of
CTI ports are OOS - all ports in TPG
3751 11:45:48.558 ACCN TELS 0 TAPIPortGroup: getActiveCCM() subsystemJTAPI
is not in service or partial service在指向的dial-peer提示：请使用一个出站转换规则并且描
出剥离被加前缀的星号(*)和替换重定向拨叫号码信息服务(RDNIS)回到原始3个数字，例如，
201，或者用全双工E.164编号，万一电话号码配置与全双工DID在提示里面。保证发送对提示
匹配为在提示侧的用户配置的电话号码邀请的转换报头。

```

波特注册故障排除的清单

1. **检查在CUCM侧的配置**：CTI Manager、CallManager和管理XML (AXL)网站服务启用并且开始？CTI端口和路由点配置并且分配唯一DN？JTAPI用户CTI启用，并且是否得以进入AXL API？JTAPI用户是否有所有CTI路由点和端口控制？有时，在配置被添加后，它是一个好想法重新启动在所有服务器的CTI Manager服务。然而，这可能是影响服务的和其可行安排维护窗口，因为这影响以CUCM使用CTI和JTAPI，例如Unified Contact Center Express的所有其它设备(UCCX)，IP管理器助理(IPMA)，Attendant Console、第三方AA或者ACD应用程序，等等。
2. **检查在提示侧的配置**：call-agent定义作为CUCM？端口许可证启用？评估许可证为初始配置是可接受。能ping CUCM？JTAPI用户凭证适当地被添加了，并且call-agent定义？模块重新加载，以便配置更改应用？如果CTI RP和端口没有从CUCM自动地导入，则请设法手工添加端口Dns在ccn子系统JTAPI下，以及每个CTI-RP的JTAPI触发和重新加载模块。

如果所有这些项目被确认，则您的下一步是获取在提示的JTAPI跟踪和CUCM CTI跟踪为了进一步可能查出问题。

相关信息

- [设置和聚集在CUE中的跟踪数据](#)
- [排除故障Cisco Unity Express系统特性](#)
- [Cisco Unity Express 的 CallManager 配置示例](#)
- [CUCM语音邮件配置文件配置](#)
- [Cisco Unity Express部署模型](#)
- [部署的Cisco Unity Express最佳实践](#)
- [CUCM转码](#)
- [CUCM MTP](#)
- [技术支持和文档 - Cisco Systems](#)

本文档是否是有用？[有](#) [没有](#)

感谢您的反馈。

[打开支持案例](#)（需要[思科服务合同](#)。）

相关的思科支持社区讨论

[思科支持社区](#)是提出和解答问题、分享建议以及与同行协作的论坛。

有关本文档中所用的规则信息，请参阅 [Cisco Technical Tips Conventions](#)。

已更新：九月06，2013

文档ID116060