

CUE JTAPI - Études de cas

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Présentation de CUE JTAPI Intégration avec CUCM](#)

[Exemple de flux d'appels de haut niveau](#)

[Activation et collecte de traces](#)

[Traces CCN JTAPI en temps réel](#)

[Journaux de suivi CCN JTAPI](#)

[Collecter les fichiers journaux de suivi](#)

[Détails obligatoires avant de vérifier les journaux](#)

[Concepts CTI de base](#)

[États d'appel CTI courants](#)

[À quoi doivent ressembler les journaux de suivi](#)

[Enregistrement du RP CTI et des ports](#)

[Appel de base transféré vers la messagerie vocale](#)

[Nouvel appel et redirection vers le port disponible](#)

[Nouvel appel et redirection vers le port disponible](#)

[Nouvel appel au port CTI](#)

[Le port CTI accepte les appels redirigés](#)

[Négociation des médias](#)

[Déconnexion d'appel](#)

[Signalisation MWI activée/désactivée](#)

[CUE active la lampe MWI pour la ligne 3001](#)

[Numéro DTMF composé '3' pour supprimer le message de la boîte aux lettres](#)

[CUE désactive la lampe MWI pour la ligne 3001](#)

[Journaux CCN en temps réel](#)

[Configuration des appels](#)

[Déconnexion d'appel](#)

[Études de cas de dépannage](#)

[Problèmes de connectivité](#)

[Utilisateur non compatible CTI](#)

[Le service CUCM CTI Manager est en panne](#)

[Incompatibilité de configuration](#)

[Problème de routage des appels CUCM](#)

[Liste de contrôle pour le dépannage de l'enregistrement des ports](#)

[Informations connexes](#)

Introduction

Ce document fournit des informations sur le dépannage de l'interface JTAPI (Java Telephony Application Programming Interface) de Cisco Unity Express (CUE). En outre, ce document fournit les informations et les commandes sur la façon d'activer, de collecter et d'afficher les différents traces et journaux avec des exemples d'études de cas de dépannage.

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Connaissances de base sur la configuration et l'utilisation de Cisco Unified Communications Manager (CUCM) via l'interface d'administration Web.
- Connaissance de base des ports et des points de routage CTI (Computer Telephony Interface) dans CUCM.
- Connaissance de base de l'interface de ligne de commande Cisco Unity Express.

Components Used

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Cisco Unity Express version 3.x ou ultérieure.
- Cisco Unified Communications Manager version 7.x ou ultérieure.

La méthode d'intégration utilisée s'applique uniquement à Cisco Unity Express avec Cisco Unified Communications Manager ; pas avec Cisco Unified Communications Manager Express (CUCME).

Cisco Unity Express doit être sous licence pour CUCM, pas CUCME. CUE peut être intégré à CUCM ou CUCME à tout moment et sous licence en conséquence.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

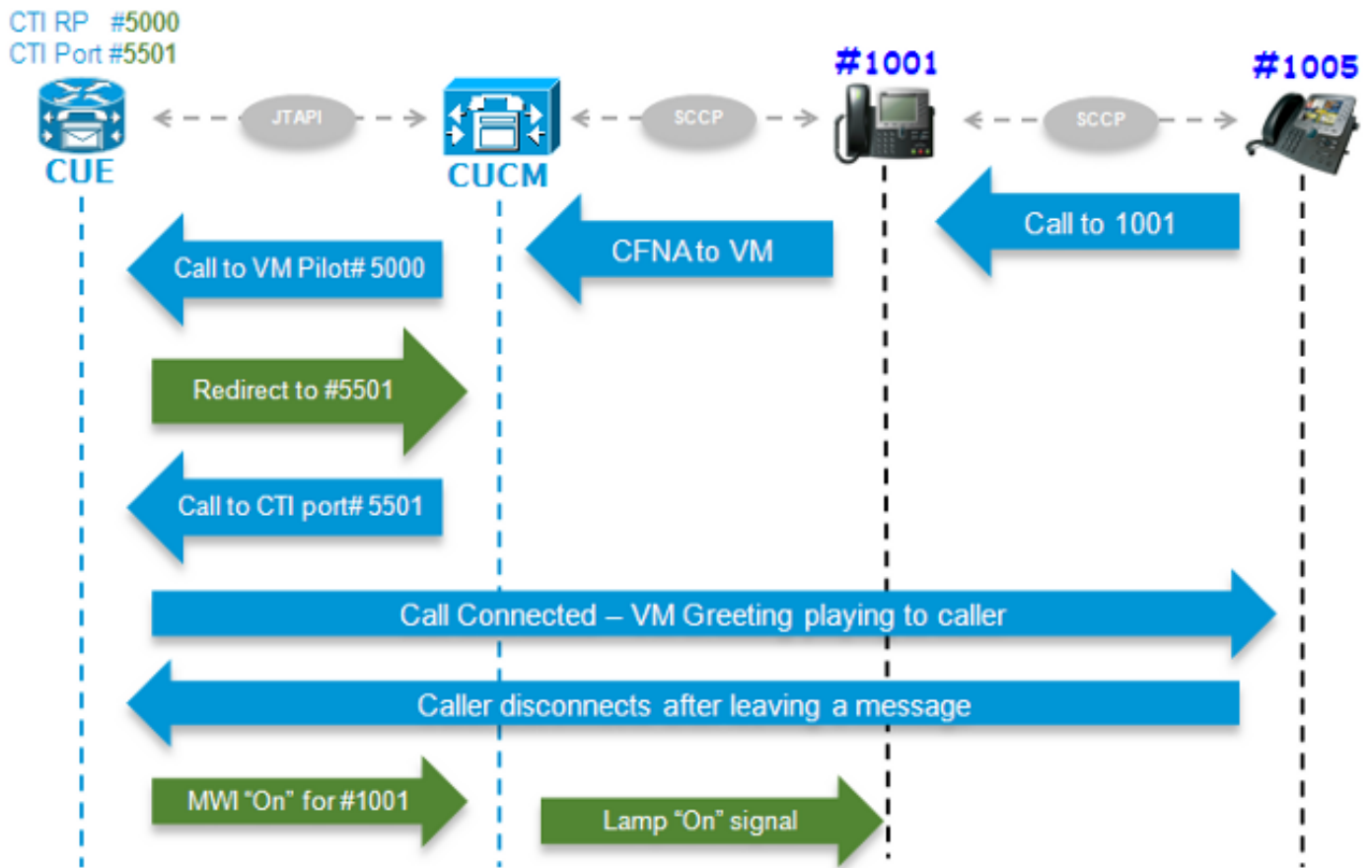
Présentation de CUE JTAPI Integration avec CUCM

Il est possible d'intégrer CUE à CUCM via le protocole JTAPI pour la messagerie vocale (VM) et la fonctionnalité de réception automatique (AA). Cette solution est recommandée lorsque vous souhaitez provisionner des fonctionnalités de VM et/ou la gestion d'appels AA de base pour un ou plusieurs sites de succursales avec un petit nombre d'utilisateurs enregistrés sur un serveur CUCM. Cela ne nécessite pas de serveur de messagerie vocale Cisco Unity complet, mais une mise en oeuvre beaucoup plus abordable. Parallèlement, la CUE fournit également des options de survie pour ses filiales et bascule vers le protocole SIP (Session Initiated Protocol) lorsque la connectivité au CUCM est perdue.

Le CUE peut s'enregistrer auprès du CUCM via JTAPI et contrôle les points de routage CTI et les ports CTI. Cela vous permet de contrôler et de gérer le CUE en tant que point de terminaison supplémentaire via le CUCM, ainsi que de faciliter les configurations et les interactions avec

d'autres points de terminaison du cluster.

Exemple de flux d'appels de haut niveau



L'utilisateur final avec le numéro de répertoire (DN) 1005 appelle l'utilisateur avec le DN 1001. L'appel est transféré au bout de quelques secondes si l'appel n'a pas obtenu de réponse (CFNA) au numéro de machine virtuelle configuré sur le profil de machine virtuelle 1001 de l'utilisateur. Le CUCM envoie ensuite l'appel au VM Pilot 5000 configuré, qui pointe vers un RP CTI avec DN 5000 contrôlé par CUE. L'application de VM CUE est déclenchée et l'appel est redirigé via JTAPI vers un port CTI disponible (DN 5501) pour l'établissement du support. Le message d'accueil audio est diffusé et l'utilisateur peut laisser un message ou interagir avec le système via des tonalités DTMF (Dual Tone Multi-Frequency). Lorsque l'appelant met fin à l'appel, le CUE indique au CUCM de définir la lampe MWI (Message Waiting Indicator) du poste 1001 sur "On" via JTAPI. Le CUCM envoie ensuite le message SCCP (Skinny Client Control Protocol) pour allumer la lumière sur le téléphone, ainsi que pour afficher l'indication d'enveloppe sur l'écran, de sorte que l'utilisateur 1001 est conscient qu'il existe un nouveau message de machine virtuelle dans la boîte aux lettres.

Activation et collecte de traces

Il existe deux types de traces :

- Suivi JTAPI Cisco Communications Network (CCN) en temps réel
- Journaux de suivi CCN JTAPI

Traces CCN JTAPI en temps réel

- Traces CCN JTAPI en temps réel. (L'activation de ces traces ne nécessite pas de rechargement du module CUE.)
- Le résultat n'est pas aussi important que les journaux de suivi CCN, mais ils ne sont pas très instructifs non plus.

Entrez ces commandes afin d'activer les traces :

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

Entrez cette commande afin de vérifier qu'ils sont activés :

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

Entrez cette commande afin de collecter le résultat :

```
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 !!
```

Entrez **CTRL-C** pour arrêter la journalisation en temps réel sur la console.

Journaux de suivi CCN JTAPI

Un rechargement du module CUE est nécessaire une fois que les journaux de suivi CCN JTAPI sont activés pour que les journaux soient remplis. Ces journaux, **messages.log** et **atrace.log**, peuvent être très détaillés ou cryptiques ainsi que beaucoup plus informatifs et détaillés. Il existe quatre journaux différents :

- **atrace.log**
Activé par défaut sur les modules de réseau (NM), mais désactivé par défaut pour les modules AIM (Advanced Integration Modules). Entrez la commande **log trace local enable** afin d'activer. Il écrit jusqu'à 10 Mo localement ou sur un serveur FTP. Afin de redémarrer le journal, entrez la commande **log trace local disable** ou la commande **no log trace local enable** ; ensuite, entrez la commande **log trace local enable**. Entrez la commande **clear trace file** afin d'effacer le **fichier atrace.log**. Les données doivent être décodées par le Centre d'assistance technique (TAC).
- **messages.log**
Il s'agit de journaux qui contiennent des messages Syslog, tels que Info, Warning, Error et Fatal.
- **CiscoJtapi1.log** et **CiscoJtapi2.log**
Ils consignent tous les événements et la signalisation JTAPI. Ces journaux sont beaucoup plus faciles à comprendre et très instructifs. Le fichier CiscoJtapi2.log commence à être renseigné lorsque CiscoJtapi1.log devient complet et vice versa.

Quelles que soient les traces définies, le système revient aux niveaux de trace par défaut après un rechargement. Pour modifier ces paramètres par défaut afin qu'ils survivent à un redémarrage, vous devez entrer la commande **log trace boot**. Voici la commande permettant de les activer :

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

Entrez cette commande afin de vérifier qu'ils sont activés :

```
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
```

Voici les étapes à suivre pour afficher les journaux :

1. Entrez la commande **show logs** afin d'afficher une liste des fichiers journaux stockés dans le CUE.
2. L'extension du fichier **.prev** signifie qu'il s'agit d'une sauvegarde d'un fichier de suivi plus ancien et non du fichier journal actif actuel.
3. Vous pouvez les extraire vers un serveur FTP externe.
4. Vous pouvez également afficher la sortie des messages qui sont enregistrés dans ces fichiers en temps réel à partir du moniteur de terminal de CUE.

Collecter les fichiers journaux de suivi

Extrayez les journaux vers un FTP externe avec les commandes suivantes :

```
copy log CiscoJtapi2.log url ftp://username:password@192.168.105.1/
copy log CiscoJtapi1.log url ftp://username:password@192.168.105.1/
copy log messages.log url ftp://username:password@192.168.105.1/
copy log atrace.log url ftp://username:password@192.168.105.1/
```

Affichez les journaux sur le moniteur de terminal CUE à l'aide de la commande **show log name <logname>**. Voici un exemple :

```
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>
```

Le fichier **atrace.log** est codé ; par conséquent, vous ne pouvez pas seulement l'afficher en temps réel avec la commande **show log name**.

Détails obligatoires avant de vérifier les journaux

Vous devez au moins obtenir tous les détails décrits dans le présent document à partir des appels portant sur le problème que vous dépannez afin de pouvoir facilement suivre et comprendre les traces :

- Numéro appelant
- Numéro appelé
- Numéro de redirection
- Nom de domaine et de périphérique du RP CTI
- Numéro de port CTI et nom du périphérique
- Utilisateur JTAPI
- Plage de temps des appels

Concepts CTI de base

Fournisseur : Fournisseur de services CTI. La demande établit une session CTI en **ouvrant** un fournisseur.

Utilisateur : Les applications sont associées à un utilisateur.

Périphérique: Périphérique qui s'enregistre auprès du CUCM.

Ligne : Apparence DN sur un périphérique pris en charge par CTI.

ID d'appel (callLegID) : Associé à une branche d'appel dans un appel.

Appel global (callID) : Identifie toutes les branches d'appel pour un seul appel.

États d'appel CTI courants

```
state = 1                IDLE
state = 2 OFFERING
state = 3 ACCEPTED
state = 8 CONNECTED
```

À quoi doivent ressembler les journaux de suivi

Avant de pouvoir détecter une signalisation incorrecte, vous devez d'abord savoir à quoi ressemblerait cette signalisation en fonctionnement normal ; cette section présente des extraits des sorties de signalisation que vous verriez dans différents scénarios lorsqu'ils fonctionnent normalement.

Sachez également que tous les signaux de ces journaux ont été résumés pour ne montrer que les détails pertinents car ils contiennent des informations très détaillées qui sont assez fastidieuses et répétitives.

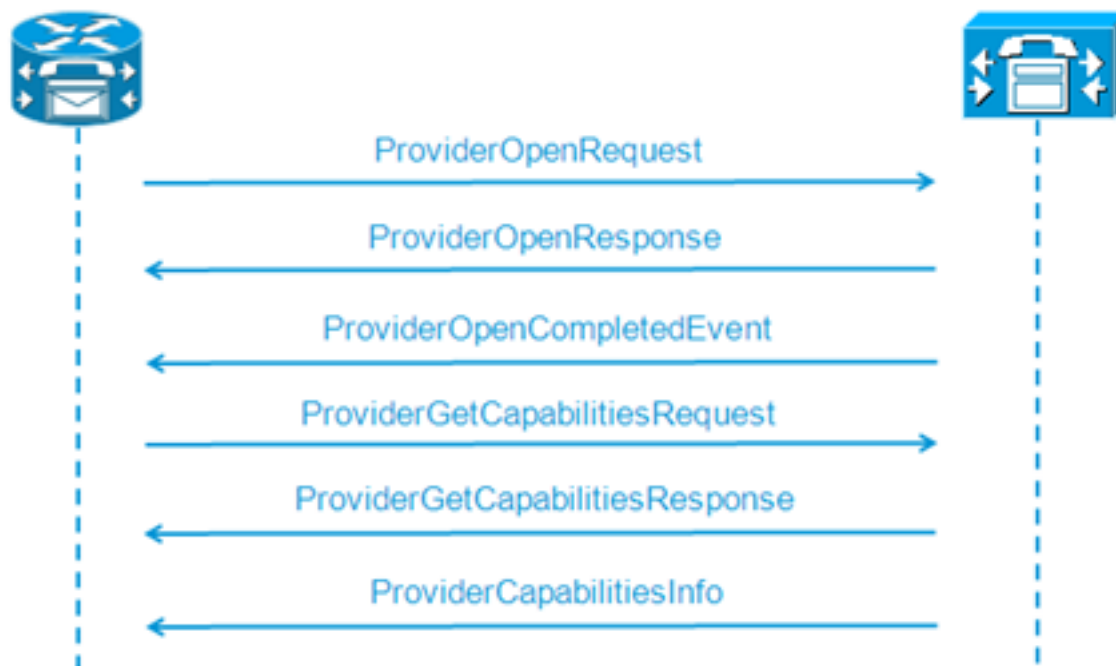
Voici les détails des configurations utilisées :

```
Jtapi User:                tacjtapiuser
CUCM IP Address: 192.168.100.10
CUE CTI Route Point: cue_vm_ctirp
CUE CTI Port: cue_ctiport1
CUE and Phone Partition: cue_pt
IP Phone MAC: SEP0023331C29EC
CTI Route Point DN: 8000
CTI Port DN: 8501
IP Phone DN: 3001
```

Enregistrement du RP CTI et des ports

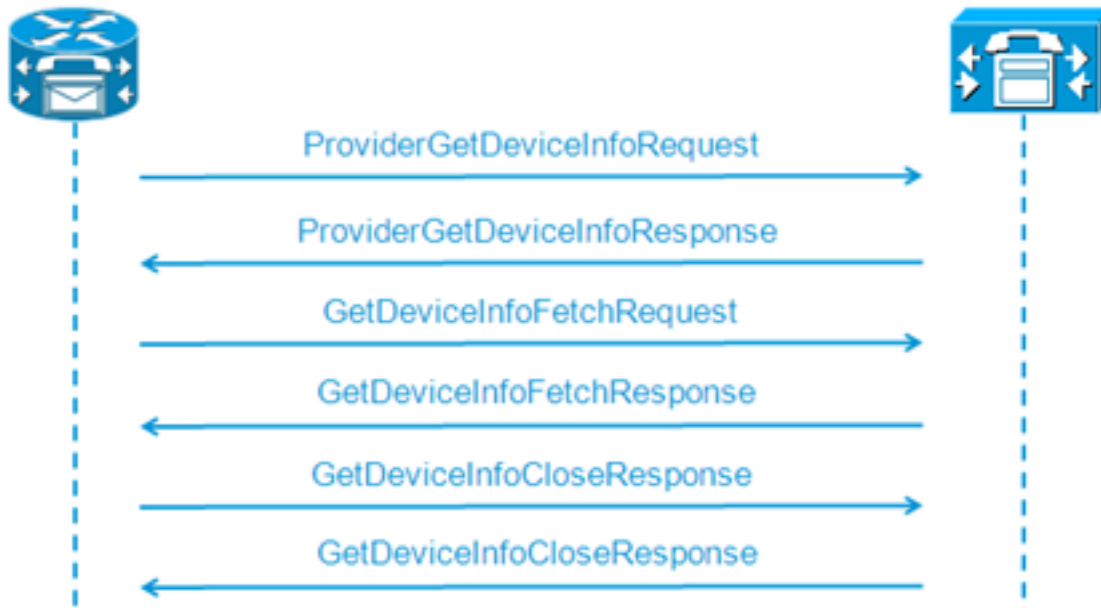
(Résultats des journaux CiscoJtapi1/Cisco Jtapi2)

1. Ouvrir une connexion fournisseur



```
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. Requête pour périphériques contrôlés



```
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. Obtenir les informations de ligne de port CTI



```
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. Obtenir les informations de ligne CTI RP

```
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. CUE applique la configuration reçue

```
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. Obtenir le contrôle des lignes et des périphériques CTI



```

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]

```

```

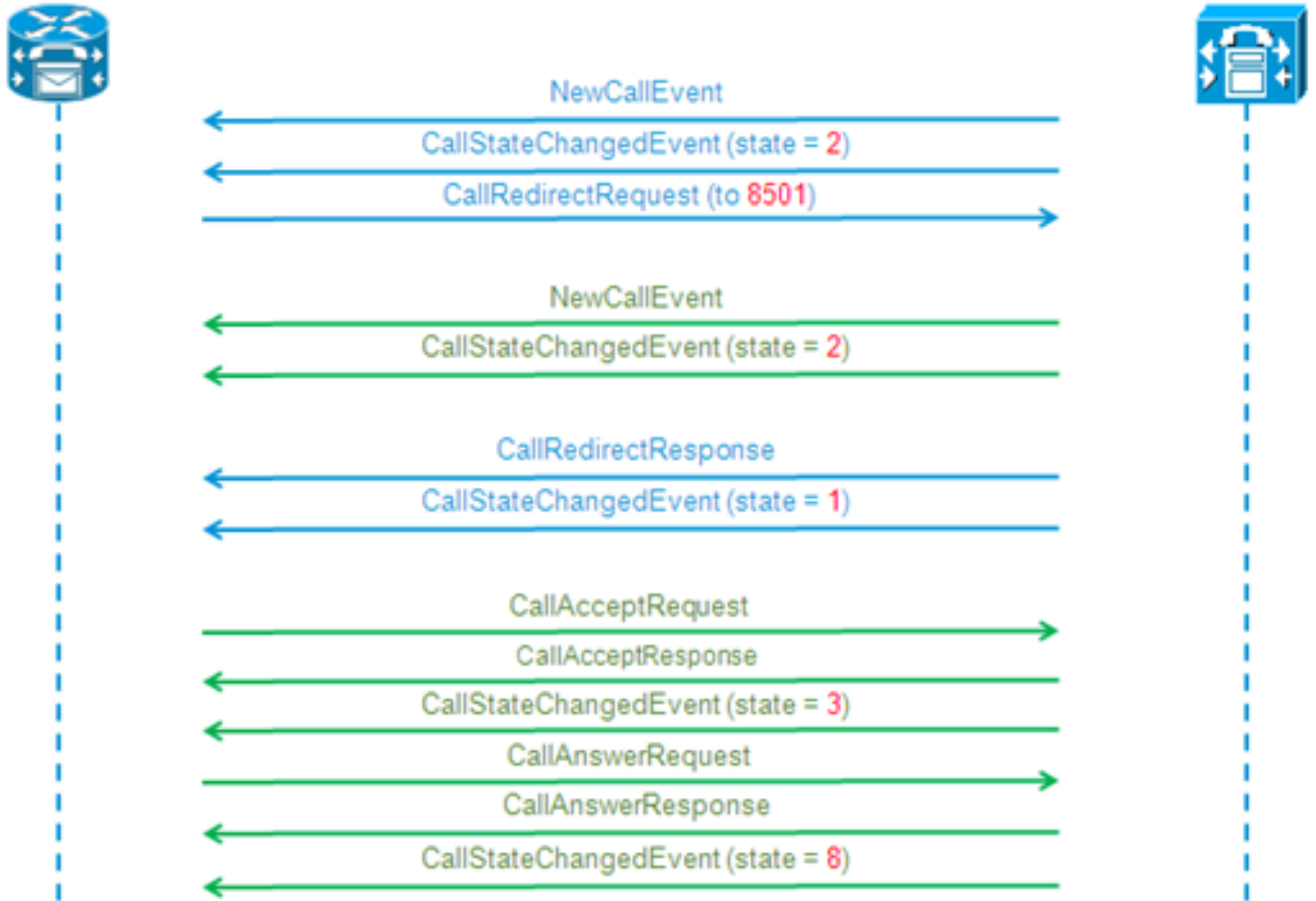
sending: com.cisco.cti.protocol.DeviceCloseRequest {
139: 12:05:25.681 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Response: com.cisco.cti.protocol.DeviceCloseResponse {
141: 12:05:25.683 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) [SS_TEL_INIT]
sending: com.cisco.cti.protocol.DeviceRegisterDeviceRequest {
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 {
}

```

Appel de base transféré vers la messagerie vocale

(Résultats des journaux CiscoJtapi1/Cisco Jtapi2)

Nouvel appel et redirection vers le port disponible



Nouvel appel et redirection vers le port disponible

```

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
}
```

Nouvel appel au port CTI

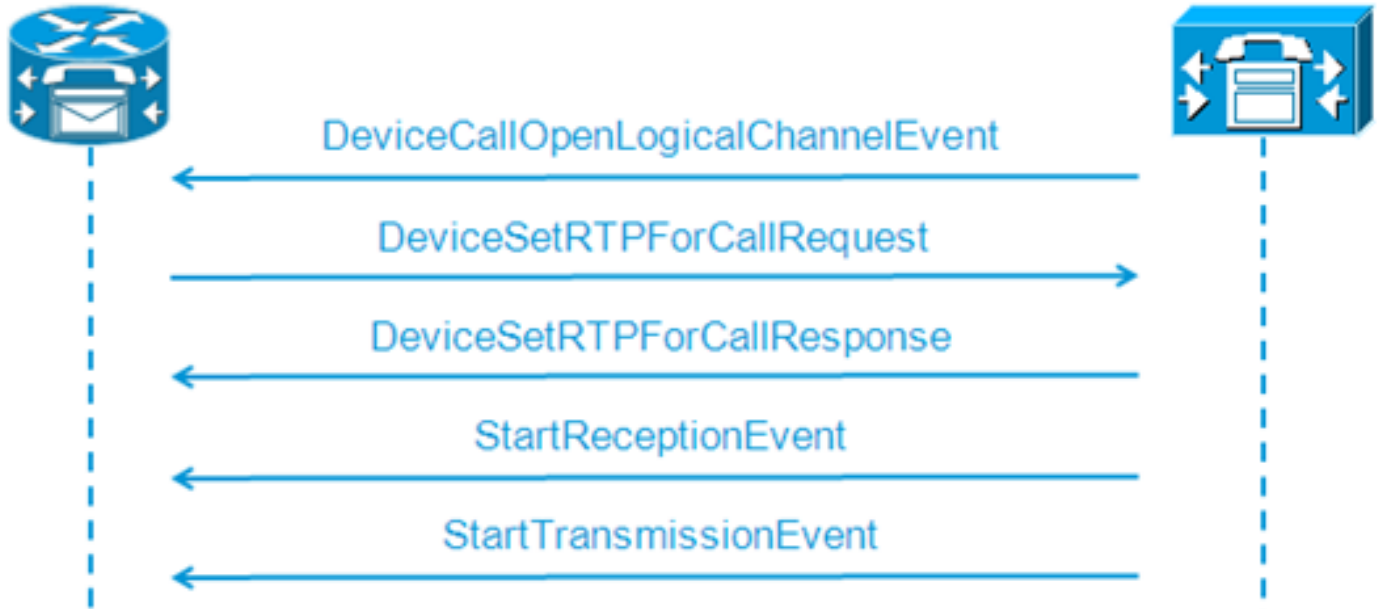
```
12:46:00.460 %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) received
Event: com.cisco.cti.protocol.NewCallEvent {
deviceName = cue_ctiport1
callLegID = 25626133
callID = 9040
callingParty = 3001
calledParty = 8501
originalCalledParty = 8000
reason = 6
lastRedirectingParty = 8000
callingPartyDeviceName = 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]
```

Le port CTI accepte les appels redirigés

```
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
```

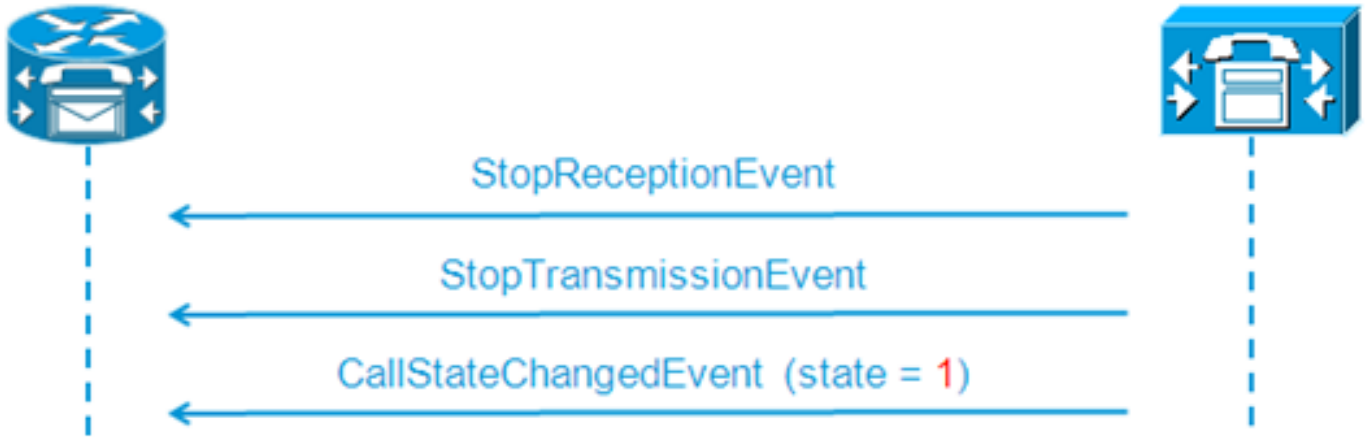
Négociation des médias



```

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
}
  
```

Déconnexion d'appel



```

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]
  
```

Signalisation MWI activée/désactivée

CUE active la lampe MWI pour la ligne 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
}
  
```

Numéro DTMF composé '3' pour supprimer le message de la boîte aux lettres

```

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]
```

CUE désactive la lampe MWI pour la ligne 3001

```
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
}
```

Journaux CCN en temps réel

(Résultats des journaux CCN en temps réel)

C'est ainsi que le même appel de l'exemple précédent apparaît ici lorsque les journaux CCN en temps réel sont collectés à la place.

Configuration des appels

```
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:41000000008 associated
with Task ID: 41000000008
12:46:00.533 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:41000000008,
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:41000000008
com.cisco.jtapi.CiscoMediaOpenLogicalChannelEvImpl received
12:46:00.593 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:41000000008
com.cisco.jtapi.CiscoRTPInputStartedEvImpl received
12:46:00.597 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:41000000008
com.cisco.jtapi.CiscoRTPOutputStartedEvImpl received
```

Déconnexion d'appel

```
12:46:09.442 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:41000000008
com.cisco.jtapi.CiscoRTPInputStoppedEvImpl received
12:46:09.443 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:41000000008
com.cisco.jtapi.CiscoRTPOutputStoppedEvImpl received
12:46:09.447 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:41000000008
gets TermConnDroppedEv, meta code:132, cause code:100
12:46:09.447 ACCN TELS 0 CallID:7 MediaId:9040/1 Task:41000000008,
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:41000000008, 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:41000000008
com.cisco.jtapi.TermObservationEndedEvImpl received
```

Études de cas de dépannage

Problèmes de connectivité

Dans ce scénario, les ports et les déclencheurs CUE ne s'enregistrent pas auprès du CUCM en raison d'un manque de connectivité entre le CUE et le CUCM.

```
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 --

```

Note: Les secondes d'horodatage vont de 13:20:28 à 13:20:38 ; par conséquent, nous pouvons dire que le CUE n'a pas pu ouvrir le socket TCP pendant 10 secondes avant d'admettre l'incapacité à créer le fournisseur.

Problèmes liés à l'authentification

Dans ce scénario, les ports et les déclencheurs CUE ne peuvent pas s'enregistrer auprès du CUCM car les mots de passe configurés entre le CUE et le CUCM ne correspondent pas.

Journal CCN

```

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

```

Utilisateur non compatible CTI

Dans ce scénario, les ports et les déclencheurs CUE ne peuvent pas s'enregistrer auprès du CUCM car l'utilisateur de l'application JTAPI n'a pas été ajouté au groupe d'autorisations **CTI standard activé** du côté CUCM. Par conséquent, même lorsque les informations d'identification de l'utilisateur s'authentifient en conséquence, l'utilisateur JTAPI, **tacjtapiuser** dans ce cas, ne peut contrôler aucun périphérique via CTI et JTAPI.

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
}

```

Le service CUCM CTI Manager est en panne

Dans ce scénario, les ports et les déclencheurs CUE ne peuvent pas s'enregistrer car le service CUCM CTI Manager est arrêté ou dans un état anormal. Il reçoit une erreur de « connexion refusée » pour la tentative de connexion de CUE au port TCP JTAPI 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
```

Incompatibilité de configuration

Dans ce scénario, le CUE ne peut pas enregistrer le déclencheur JTAPI avec le numéro 9999, car le RP CTI qu'il doit correspondre n'est pas configuré, ou il n'a pas été ajouté aux périphériques contrôlables " " pour l'utilisateur côté CUCM. Le CUE le réalise après avoir reçu la **GetDeviceInfoFetchResponse** de CUCM et remarque qu'il n'existe pas de périphérique dans le domaine du fournisseur, qui fait référence à tous les périphériques contrôlables par cet utilisateur, qui correspondraient au numéro de déclencheur qu'il a configuré localement. Le CUE n'essaie alors pas d'envoyer un **DeviceOpenRequest** pour ce déclencheur spécifique et ne signale à la place que l'exception dans les traces. Le CUE tente toujours d'enregistrer tous les autres périphériques qui se trouvent dans le domaine du fournisseur envoyé par le CUCM.

```
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.
```

Note: Même lorsque le déclencheur 9999 est configuré localement dans le CUE, il ne fait pas partie du domaine du fournisseur reçu du CUCM et ne s'enregistre donc pas.

CUE continue à ouvrir la ligne 8000 ; qui est inclus dans le domaine du fournisseur

```
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
```

Problème de routage des appels CUCM

Dans ce scénario, l'utilisateur avec DN 3001 appelle le CUE pour vérifier sa machine virtuelle. L'appel est présenté au pilote de machine virtuelle (CTI RP) de CUE avec DN 8000. Le CUE demande ensuite à l'appel d'être redirigé vers son port CTI média avec DN 8501, mais l'appel ne parvient pas à être redirigé car le CSS configuré pour DN 3001 n'a pas accès au PT où le DN du port CTI est attribué.

```
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

```

Problèmes de licence

Dans ce scénario, le CUE ne peut pas enregistrer ses ports et déclencheurs car les licences des ports de VM n'ont pas été activées. Aucune tentative d'enregistrement n'est vue dans les captures pour la même raison.

Récapitulatif tiré du fichier atrace.log décodé :

```

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():

```

```

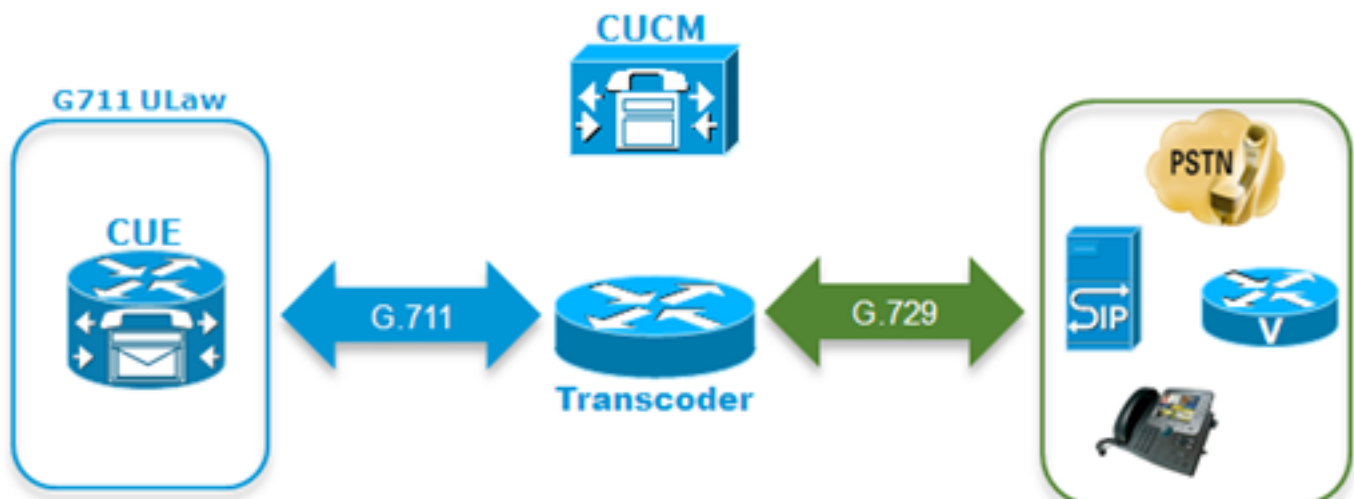
2555 11:45:23.114 LLMA LPRT 0 LlamaPortQuery: get(): maxCount
3456 11:45:23.115 LLMA LSDB 0 Llama: getMaxPortCount(): LlamaSysdbUser():
getInt(): Getting int/sw/apps/limitsManager/port/query/maxCount returns 0
3456 11:45:23.115 LLMA LAPI 0 Llama: getMaxPortCount(): count: 0
3456 11:45:28.727 ACCN TELS 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

```

Meilleures pratiques

CUE prend uniquement en charge le codec G711ulaw ; par conséquent, dans presque tous les déploiements, un transcodeur est nécessaire pour que le CUE puisse communiquer avec d'autres périphériques ou agrégations qui utilisent des codecs différents (inclut G711Alaw). Il en va de même pour les interconnexions DTMF avec des périphériques qui prennent uniquement en charge DTMF intrabande lorsqu'une ressource MTP (Media Termination Point) est également requise. En raison de ces limitations, Cisco recommande :

- Créez un pool de périphériques isolé pour utiliser uniquement les RP CTI et les ports CTI de CUE. Si plusieurs CUE sont intégrés à CUCM, créez un pool de périphériques par CUE.
- Créez une région distincte uniquement pour le RP et les ports du CUE et appliquez-la à ce pool de périphériques isolé.
- Assurez-vous que la région est configurée pour autoriser uniquement G711 avec toutes les autres régions.
- Assurez-vous qu'une liste MRGL (Media Resource Group List) avec les ressources de transcodage disponibles est appliquée au pool de périphériques des ports CTI RP et CTI de CUE afin qu'ils aient accès à une ressource de transcodage, si nécessaire.



- Si l'utilisateur ne parvient pas à naviguer dans les menus vocaux avec des tonalités DTMF, il est possible qu'une ressource MTP soit ajoutée au MRGL des périphériques CUE.

Créer un profil de machine virtuelle distinct pour l'CUE dans CUCM

Afin d'éviter certains problèmes récents observés avec le gestionnaire CTI de CUCM, il est recommandé d'associer tous les téléphones à l'utilisateur JTAPI de CUE côté CUCM, au lieu des seuls RP et ports CTI.

Si la fonctionnalité SRST (Survivable Remote Site Telephony) est souhaitée :

- Assurez-vous que le déclencheur SIP correspondant est configuré pour chaque déclencheur JTAPI sur le CUE.
- Assurez-vous que les terminaux de numérotation dial-peer sont ajoutés au routeur de la filiale afin de permettre le routage des appels vers le module CUE via SIP en mode SRST.
- Configurez le masque de numéro externe de chacun des points de routage CTI, ainsi que le masque du champ CFU (Call Forward Unregistered) dans CUCM pour s'assurer que les routes CUCM des appels destinés au module de filiale via la passerelle RTPC (Local Public Switched Phone Network) lorsque la connectivité entre CUCM et CUE a été perdue ou si le routage AAR (Automatic Alternate Routing) est appelé. Des règles de traduction supplémentaires peuvent être requises pour que le routeur de filiale puisse également acheminer les appels entrants du RTPC vers le module CUE.
- Si l'approche de configuration Transfert direct vers VM est présente du côté CUCM et que l'utilisateur souhaite conserver cette fonctionnalité pendant qu'il est dans CME-SRST, vous devez utiliser l'ancien DN factice avec l'approche de configuration Renvoi de tous les appels (CFA) qui a été utilisée pour CME avant que la touche de fonction **TransferToVM** ne soit disponible. Référez-vous à [Transférer un appelant directement dans une boîte aux lettres Unity Express](#) pour plus d'informations. Voici un exemple de la façon dont cela peut ressembler. N'oubliez pas que ceci ne peut être fait que si **CME-SRST** est utilisé et non pas SRST hérité avec call-manager-fallback. Supposez que les DN se situent dans la plage 200-299. Appel entrant pour x201. Configurez un **ephone-dn** avec cette commande :

```
ephone-dn 99
number 2..
call-forward all <VM Pilot>
```

Dans le terminal de numérotation dial-peer pointant vers CUE : Utilisez une règle et un profil de traduction sortants pour supprimer l'astérisque préfixé ("*") et remplacer le **service RDNIS (Redirect Dialed Number Information Service)** par le numéro à 3 chiffres d'origine, par exemple 201, ou par le numéro E.164 complet si le **numéro de téléphone** a été configuré avec le DID complet à l'intérieur du CUE. Assurez-vous que l'en-tête Diversion de l'INVITE envoyé au CUE correspond au **numéro de téléphone** configuré pour l'utilisateur côté CUE.

Liste de contrôle pour le dépannage de l'enregistrement des ports

1. **Vérifiez la configuration côté CUCM** : Les services Web CTI Manager, CallManager et Administrative XML (AXL) sont-ils activés et démarrés ? Les ports CTI et les points de routage ont-ils été configurés et ont-ils reçu un DN unique ? L'utilisateur JTAPI CTI est-il activé et dispose-t-il d'un accès à l'API AXL ? L'utilisateur JTAPI contrôle-t-il tous les points de routage et ports CTI ? Il est parfois judicieux de redémarrer le service CTI Manager sur tous les serveurs après l'ajout de la configuration. Cependant, il peut s'agir d'un impact sur le service et il est conseillé de planifier une fenêtre de maintenance, car cela affecte tous les autres périphériques qui utilisent CTI et JTAPI avec CUCM, tels que Unified Contact Center

Express (UCCX), IP Manager Assistant (IPMA), Attendant Console, applications tierces AA ou ACD, etc.

2. **Vérifiez la configuration côté CUE** : L'agent d'appel est-il défini comme CUCM ? Les licences de port ont-elles été activées ? Les licences d'évaluation sont acceptables pour la configuration initiale. Pouvez-vous envoyer une requête ping au CUCM ? Les informations d'identification de l'utilisateur JTAPI ont-elles été ajoutées correctement et les **agents d'appel ont-ils** été définis ? Le module a-t-il été rechargé afin que les modifications de configuration soient appliquées ? Si le RP CTI et le port ne sont pas importés automatiquement à partir de CUCM, essayez d'ajouter manuellement les DN de port sous **jtapi du sous-système ccn**, ainsi que les déclencheurs **jtapi** pour chaque CTI-RP et rechargez le module.

Si tous ces éléments sont confirmés, alors votre prochaine étape est d'obtenir des traces JTAPI sur les traces CUE et éventuellement CUCM CTI afin d'isoler davantage le problème.

Informations connexes

- [Configurer et collecter des données de suivi dans CUE](#)
- [Dépannage des fonctionnalités du système Cisco Unity Express](#)
- [Exemple de configuration de CallManager pour Cisco Unity Express](#)
- [Configuration du profil de messagerie vocale CUCM](#)
- [Modèles de déploiement Cisco Unity Express](#)
- [Meilleures pratiques de déploiement de Cisco Unity Express](#)
- [Transcodage CUCM](#)
- [MTP CUCM](#)
- [Support et documentation techniques - Cisco Systems](#)