

CUE JTAPI-Probleme und Anwenderberichte

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[CUE JTAPI-Integration mit CUCM - Übersicht](#)

[High-Level Call Flow - Beispiel](#)

[Aktivierung und Erfassung von Ablaufverfolgungen](#)

[Echtzeit-JTAPI CCN Traces](#)

[JTAPI CCN Trace-Protokolle](#)

[Erfassen der Trace-Protokolldateien](#)

[Vor dem Überprüfen der Protokolle müssen Details vorhanden sein.](#)

[Grundlegende CTI-Konzepte](#)

[Häufige CTI-Anrufstatus](#)

[Wie Trace-Protokolle aussehen sollten](#)

[CTI RP und Portregistrierung](#)

[Einfache Weiterleitung von Anrufen an Voicemail](#)

[Neuer Anruf und Umleitung an verfügbaren Port](#)

[Neuer Anruf und Umleitung an verfügbaren Port](#)

[Neuer Anruf an CTI-Port](#)

[CTI-Port akzeptiert umgeleiteten Anruf](#)

[Medienverhandlungen](#)

[Anruftrennung](#)

[MWI-Ein/Aus-Signalisierung](#)

[CUE schaltet MWI-Lampe für Leitung 3001 ein](#)

[Gewählte DTMF-Nummer '3' zum Löschen der Nachricht aus dem Postfach](#)

[CUE schaltet MWI-Lampe für Leitung 3001 aus](#)

[Echtzeit-CCN-Protokolle](#)

[Anrufeinrichtung](#)

[Anruftrennung](#)

[Problembehebung - Anwenderberichte](#)

[Verbindungsprobleme](#)

[Benutzer nicht CTI-fähig](#)

[Der CUCM CTI Manager-Dienst ist ausgefallen.](#)

[Konfigurationsfehler](#)

[CUCM-Anrufweiterleitungsproblem](#)

[Checkliste für die Portregistrierung Fehlerbehebung](#)

[Zugehörige Informationen](#)

Einführung

Dieses Dokument enthält Informationen zur Fehlerbehebung bei der Java Telephony Application Programming Interface (JTAPI) von Cisco Unity Express (CUE). Darüber hinaus enthält dieses Dokument Informationen und Befehle zum Aktivieren, Erfassen und Anzeigen der verschiedenen Ablaufverfolgungen und Protokolle sowie Beispiele von Problemlösungsbeispielen.

Voraussetzungen

Anforderungen

Cisco empfiehlt, über Kenntnisse in folgenden Bereichen zu verfügen:

- Grundkenntnisse der Konfiguration und Verwendung von Cisco Unified Communications Manager (CUCM) über die Webverwaltungsschnittstelle.
- Grundkenntnisse der CTI-Ports (Computer Telephony Interface) und Route Points (RPs) in CUCM.
- Grundlegende Kenntnisse der Befehlszeilenschnittstelle von Cisco Unity Express.

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf den folgenden Software- und Hardwareversionen:

- Cisco Unity Express Version 3.x oder höher
- Cisco Unified Communications Manager 7.x oder höher

Die verwendete Integrationsmethode gilt nur für Cisco Unity Express mit Cisco Unified Communications Manager. nicht mit Cisco Unified Communications Manager Express (CUCME).

Cisco Unity Express muss für CUCM lizenziert sein, nicht für CUCME. CUE kann jederzeit in CUCM oder CUCME integriert und entsprechend lizenziert werden.

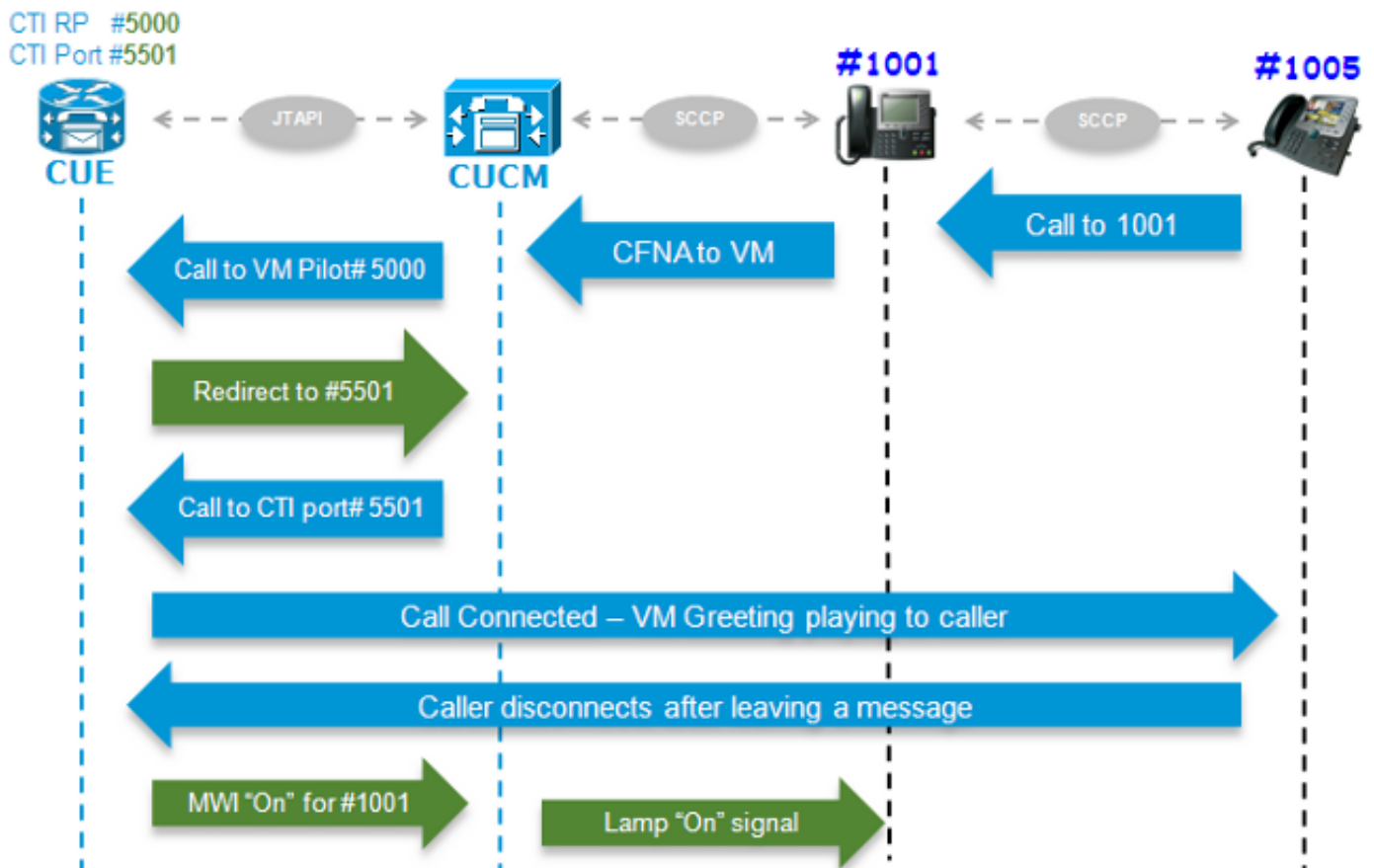
Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

CUE JTAPI-Integration mit CUCM - Übersicht

CUE kann über das JTAPI-Protokoll für Voicemail (VM) und die automatische Anrufvermittlung (AA) in CUCM integriert werden. Diese Lösung wird empfohlen, wenn Sie VM-Funktionen und/oder eine grundlegende AA-Anrufbearbeitung für eine oder mehrere Zweigstellen mit einer geringen Anzahl von Benutzern bereitstellen möchten, die auf einem CUCM-Server registriert sind. Hierfür ist kein vollwertiger Cisco Unity Voicemail-Server erforderlich, sondern eine deutlich erschwinglichere Implementierung. Gleichzeitig bietet CUE auch Optionen für die Ausfallsicherheit für die Zweigstellen und ermöglicht den Failover auf das Session Initiated Protocol (SIP), wenn die Verbindung zum CUCM unterbrochen wird.

CUE kann sich über JTAPI beim CUCM registrieren lassen und kontrolliert CTI-Routenpunkte und CTI-Ports. So können Sie CUE über den CUCM als zusätzlichen Endpunkt steuern und verwalten sowie Konfigurationen und Interaktionen mit anderen Endpunkten im Cluster vereinfachen.

High-Level Call Flow - Beispiel



Der Endbenutzer mit Verzeichnisnummer (DN) 1005 ruft den Benutzer mit DN 1001 an. Der Anruf wird nach ein paar Sekunden weitergeleitet, wenn der Anruf nicht entgegengenommen wird, Call Forward No Answer (CFNA), an die im VM-Profil des Benutzers 1001 konfigurierte VM-Nummer. Der CUCM sendet den Anruf dann an den konfigurierten VM Pilot 5000, der auf einen CTI-RP mit DN 5000 zeigt, der von CUE gesteuert wird. Die CUE VM-Anwendung wird ausgelöst, und der Anruf wird über JTAPI an einen verfügbaren CTI-Port (DN 5501) für die Medieneinrichtung umgeleitet. Die Audiobegrüßung wird wiedergegeben, und der Benutzer kann eine Nachricht hinterlassen oder über DTMF-Töne (Dual Tone Multi Frequency) mit dem System interagieren. Wenn der Anrufer den Anruf beendet, signalisiert CUE dem CUCM, die MWI-Lampe (Message Waiting Indicator) für die Durchwahl 1001 auf "On" durch JTAPI einzustellen. Der CUCM sendet dann die SCCP-Nachricht (Skinny Client Control Protocol), um die Lampe am Telefon einzuschalten, und zeigt die Umschlaganzeige auf dem Display an, sodass Benutzer 1001 weiß, dass eine neue VM-Nachricht in der Mailbox vorhanden ist.

Aktivierung und Erfassung von Ablaufverfolgungen

Es gibt zwei Arten von Ablaufverfolgungen:

- Echtzeit-JTAPI Cisco Communications Network (CCN)-Traces
- JTAPI CCN-Ablaufverfolgungsprotokolle

Echtzeit-JTAPI CCN Traces

- Echtzeit-JTAPI CCN-Traces. (Um diese Traces zu aktivieren, muss das CUE-Modul nicht neu geladen werden.)

- Die Ausgabe ist nicht so umfangreich wie die CCN-Ablaufverfolgungsprotokolle, aber auch nicht sehr informativ.

Geben Sie die folgenden Befehle ein, um die Ablaufverfolgungen zu aktivieren:

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

Geben Sie diesen Befehl ein, um zu überprüfen, ob diese aktiviert sind:

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

Geben Sie diesen Befehl ein, um die Ausgabe zu erfassen:

```
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 !!
```

Geben Sie **STRG-C** ein, um die Echtzeitprotokollierung an der Konsole zu stoppen.

JTAPI CCN Trace-Protokolle

Nach Aktivierung der JTAPI CCN-Ablaufverfolgungsprotokolle muss das CUE-Modul neu geladen werden, damit die Protokolle aufgefüllt werden können. Diese Protokolle, **messages.log** und **atrace.log**, können sehr detailliert oder kryptisch sein, sowie viel informativer und detaillierter. Es gibt vier verschiedene Protokolle:

- **atrace.log**
Standardmäßig auf Netzwerkmodulen aktiviert, aber standardmäßig deaktiviert für AIMS (Advanced Integration Modules). Geben Sie den Befehl **log trace local enable** ein, um zu aktivieren. Es schreibt bis zu 10 MB lokal oder auf einen FTP-Server. Um das Protokoll neu zu starten, geben Sie den Befehl **log trace local disable** oder den Befehl **no log trace local enable** ein. Geben Sie dann den Befehl **log trace local enable** ein. Geben Sie den Befehl **clear trace file** ein, um die Datei **atrace.log** zu löschen. Die Daten müssen vom Technical Assistance Center (TAC) dekodiert werden.
- **messages.log**
Dies sind Protokolle, die Syslog-Meldungen wie Info, Warning, Error und Fatal enthalten.
- **CiscoJtapi1.log** und **CiscoJtapi2.log**
Sie protokollieren alle JTAPI-bezogenen Signalisierungen und Ereignisse. Diese Protokolle sind viel einfacher zu verstehen und sehr informativ. Die Datei **CiscoJtapi2.log** wird angezeigt, wenn **CiscoJtapi1.log** voll ist und umgekehrt.

Unabhängig davon, welche Ablaufverfolgungen festgelegt werden, kehrt das System nach dem erneuten Laden auf die Standard-Ablaufverfolgungsebenen zurück. Um diese Standardeinstellungen so zu ändern, dass sie einen Neustart überstehen, müssen Sie den Befehl **log trace boot** eingeben. Mit dem folgenden Befehl können Sie sie aktivieren:

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

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

Geben Sie diesen Befehl ein, um zu überprüfen, ob diese aktiviert sind:

```
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
```

So zeigen Sie die Protokolle an:

1. Geben Sie den Befehl **show logs (Protokolle anzeigen)** ein, um eine Liste der in CUE gespeicherten Protokolldateien anzuzeigen.
2. Die Dateierweiterung **.prev** bedeutet, dass es sich um eine Sicherung einer älteren Ablaufverfolgungsdatei und nicht um die aktuelle aktive Protokolldatei handelt.
3. Sie können sie auf einen externen FTP-Server extrahieren.
4. Sie können die Ausgabe der Nachrichten, die an diesen Dateien protokolliert werden, auch in Echtzeit vom Terminalmonitor von CUE anzeigen.

Erfassen der Trace-Protokolldateien

Extrahieren Sie die Protokolle mithilfe der folgenden Befehle auf ein externes FTP:

```
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/
```

Anzeigen von Protokollen auf dem CUE-Terminalmonitor mit dem Befehl **show log name <logname>**. Hier ein Beispiel:

```
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>
```

Die Datei **atrace.log** ist kodiert. Sie können es daher nicht nur in Echtzeit mit dem Befehl **show log name** anzeigen.

Vor dem Überprüfen der Protokolle müssen Details vorhanden sein.

Sie müssen mindestens alle hierin aufgeführten Details aus den Anrufen mit dem Problem beziehen, das Sie beheben möchten, damit Sie die Spuren leicht verfolgen und verstehen können:

- Anrufernummer
- Angerufene Nummer

- Umleitungsnummer
- CTI RP-DN und Gerätename
- CTI-Portnummer und Gerätename
- JTAPI-Benutzer
- Zeitbereich der Anrufe

Grundlegende CTI-Konzepte

Anbieter: Ein Anbieter von CTI-Services. Die Anwendung erstellt eine CTI-Sitzung, indem sie einen Anbieter **öffnet**.

Benutzer: Anwendungen sind einem Benutzer zugeordnet.

Gerät: Ein Gerät, das sich beim CUCM registriert.

Leitung: DN-Darstellung auf einem CTI-unterstützten Gerät.

Call ID (CallLegID): Zuordnung zu einem Anrufabschnitt

Globaler Anruf (Anruf-ID): Identifiziert alle Anrufabschnitte für einen einzelnen Anruf.

Häufige CTI-Anrufstatus

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

Wie Trace-Protokolle aussehen sollten

Bevor Sie eine falsche Signalisierung finden, müssen Sie zunächst wissen, wie diese Signalisierung im Normalbetrieb aussehen würde. In diesem Abschnitt werden Ausschnitte der Signalisierungsausgaben gezeigt, die Sie in verschiedenen Szenarien sehen würden, wenn sie normal funktionieren.

Bitte beachten Sie auch, dass alle Signalisierungen aus diesen Protokollen zusammengefasst wurden, um nur die relevanten Details anzuzeigen, da sie sehr detaillierte Informationen enthalten, die recht mühsam und wiederholend sind.

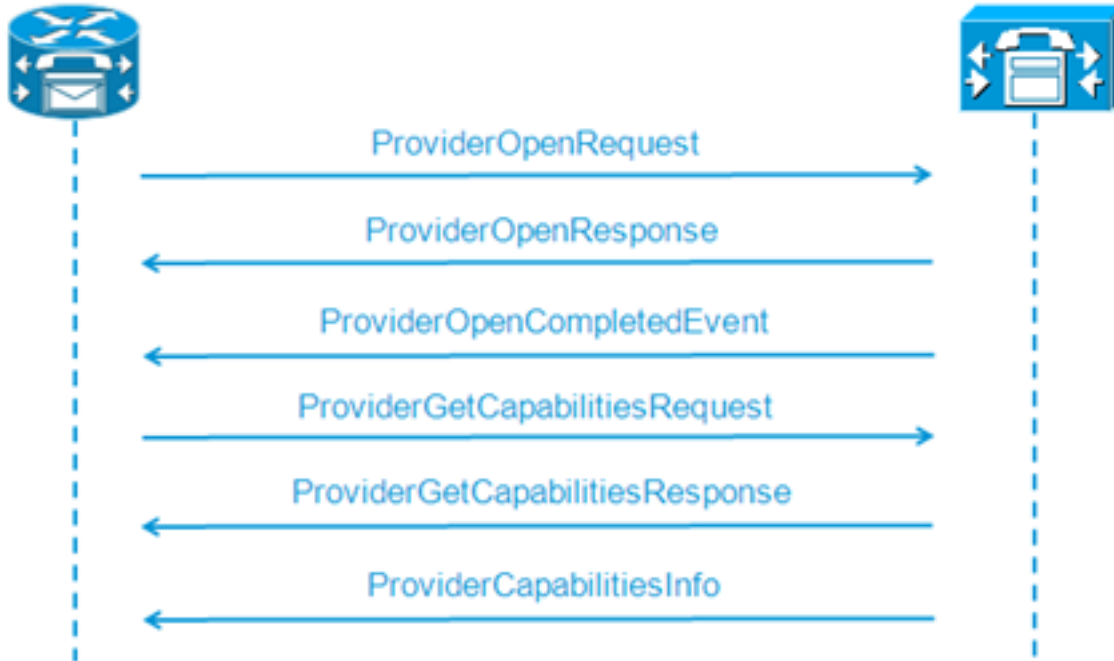
Nachfolgend sind die Details der verwendeten Konfigurationen aufgeführt:

```
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
```

CTI RP und Portregistrierung

(Ausgang aus Cisco Jtapi1/Cisco Jtapi2 Logs)

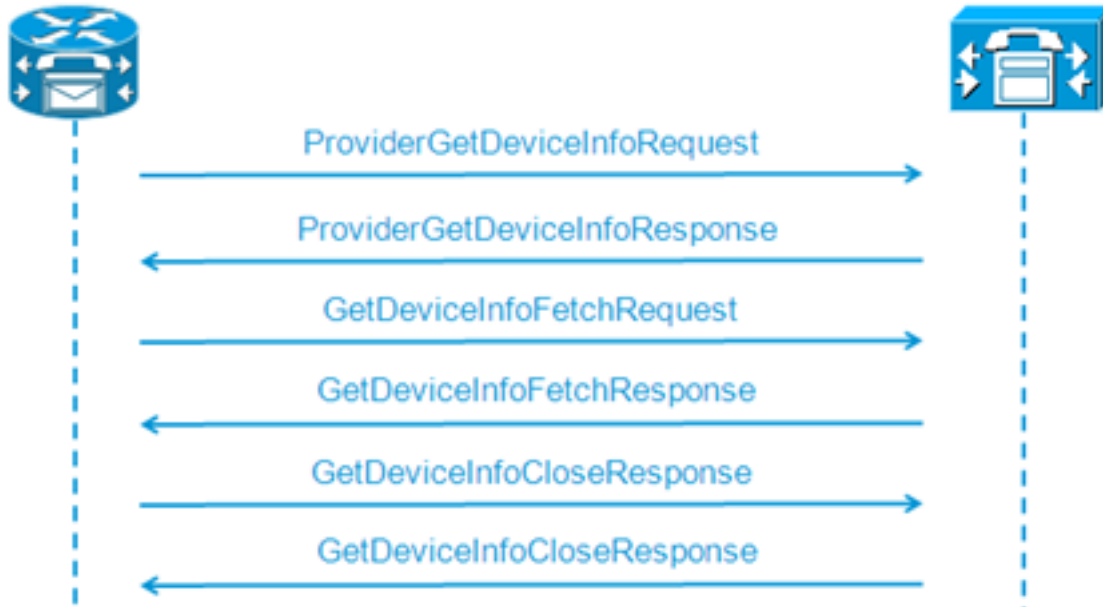
1. Provider-Verbindung öffnen



```

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
}
  
```

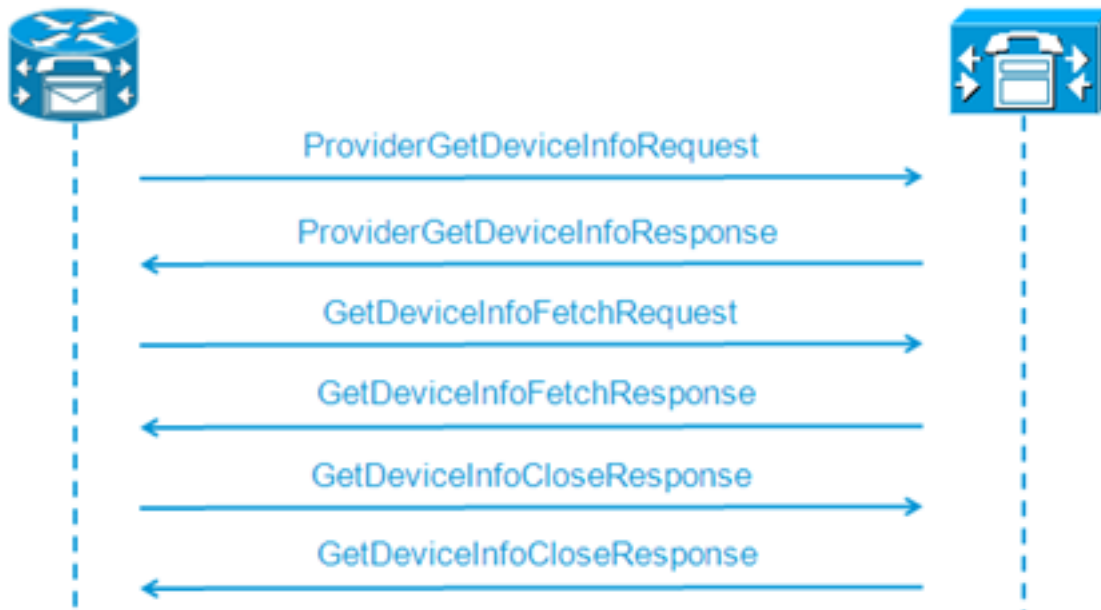
2. Abfrage nach steuerbaren Geräten



```

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)
  
```


3. Abrufen von CTI-Portleitungsinformationen



```

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. Informationen zu CTI-RP-Leitungen

```
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 wendet empfangene Konfiguration an

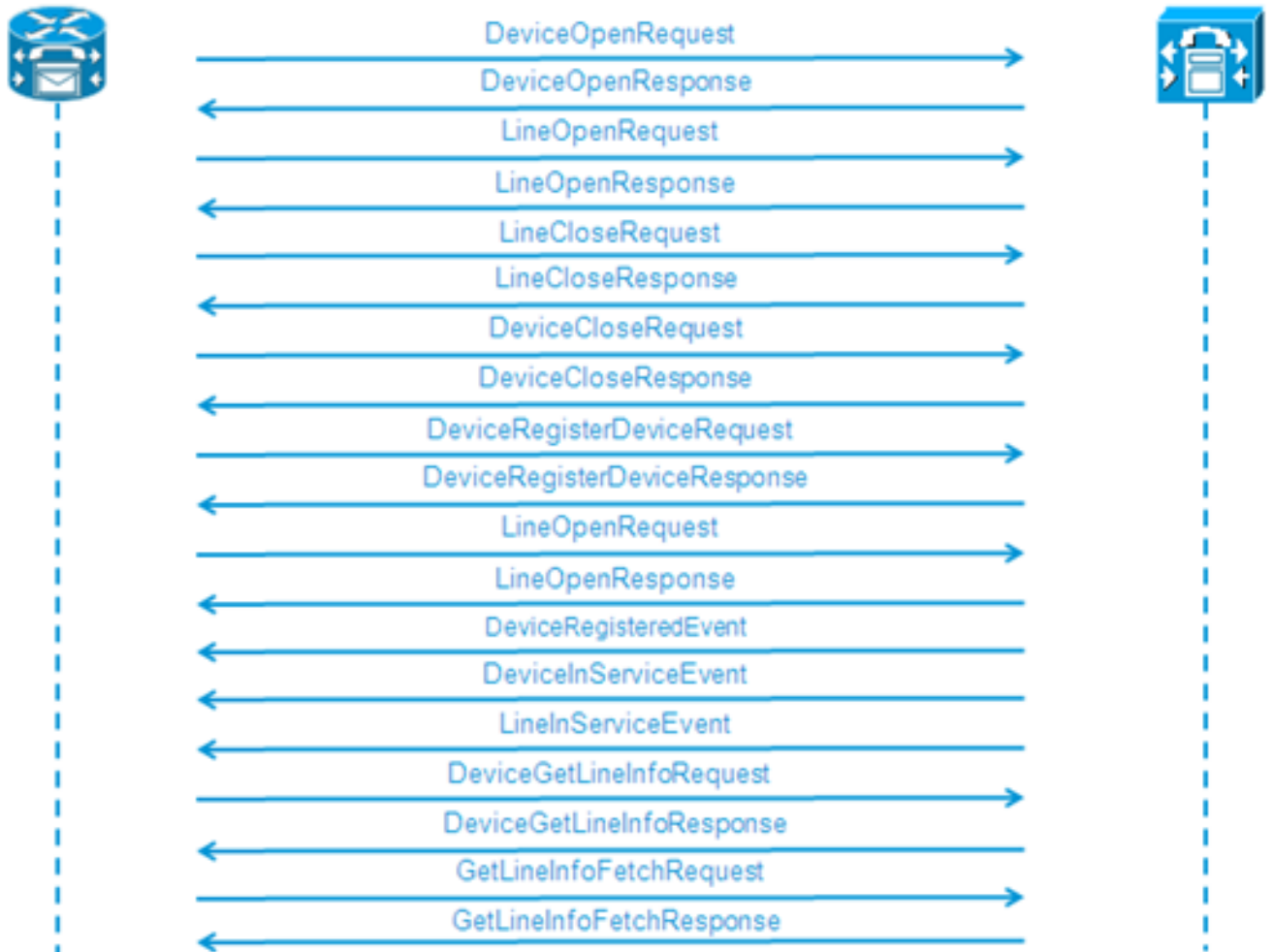
```
76: 12:05:25.064 CST %JTAPI-MISC-7-UNK.Provider 192.168.100.10
open, beginning device
initialization
77: 12:05:25.071 CST %JTAPI-JTAPI-7-UNK. (P1-tacjtapiuser) [SS_TEL_INIT]
(P1-tacjtapiuser) Request: addObserver
79: 12:05:25.073 CST %JTAPI-MISC-7-UNK.ObserverThread
(com.cisco.wf.subsystems.jtapi.SubsystemJTAPI$ProviderObserver@3d823d82):created
80:12:05:25.074 CST %JTAPI-JTAPI-7-UNK. (P1-tacjtapiuser) ProvOutOfServiceEv [#0]
Cause:100 CallCtlCause:0 CiscoFeatureReason:12
82: 12:05:25.085 CST %JTAPI-MISC-7-
UNK.ObserverThread
(com.cisco.wf.subsystems.jtapi.SubsystemJTAPI$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. Kontrolle über CTI-Geräte und -Leitungen



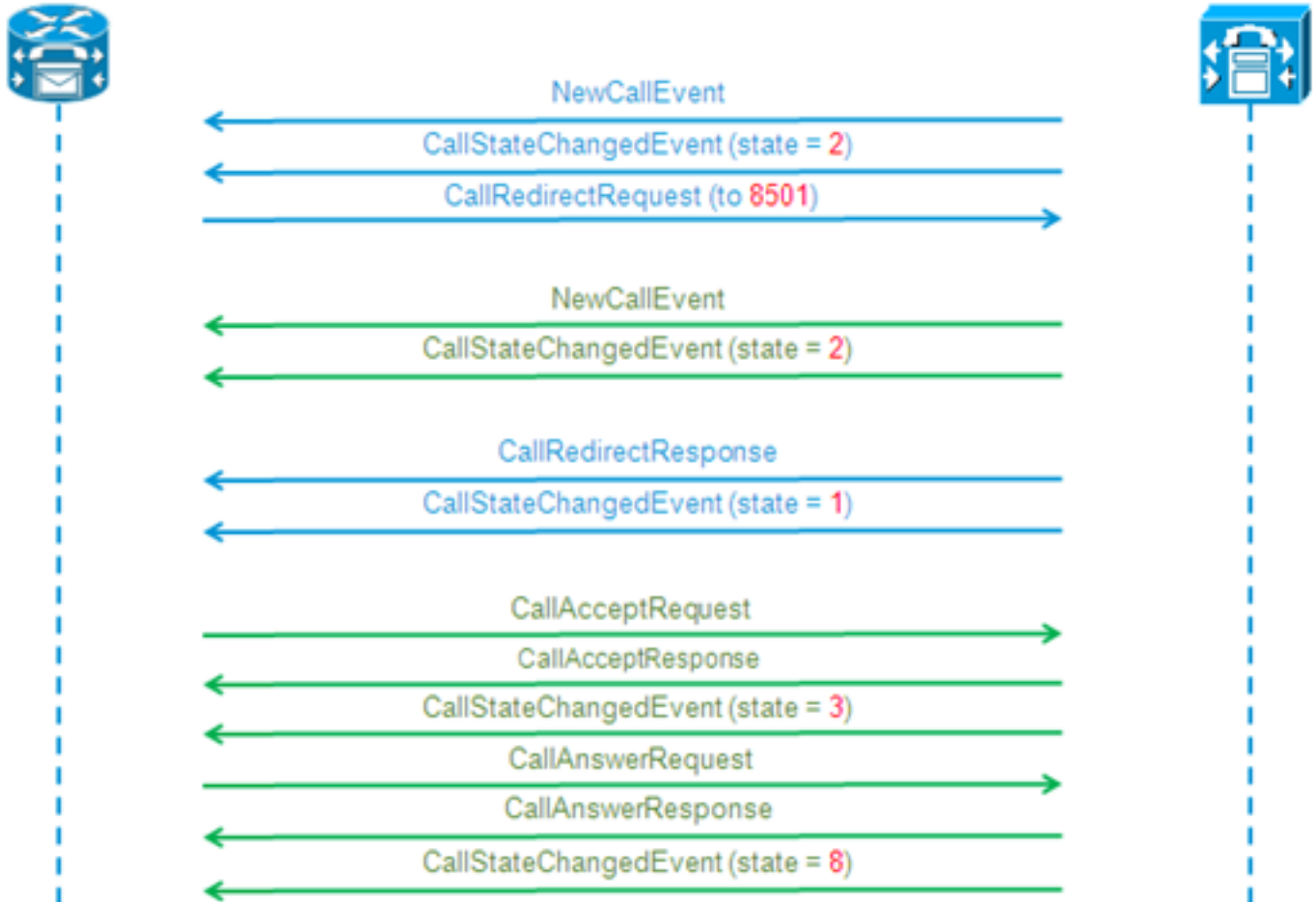
109: 12:05:25.528 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10) [SS_TEL_INIT]
sending:
com.cisco.cti.protocol.**DeviceOpenRequest** {
deviceName = **cue_ctiport1**
}
110: 12:05:25.533 CST %JTAPI-PROTOCOL-7-UNK.(P1-192.168.100.10)
received Response:
com.cisco.cti.protocol.**DeviceOpenResponse** {
result = 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 {  
}
```

Einfache Weiterleitung von Anrufen an Voicemail

(Ausgang aus Cisco Jtapi1/Cisco Jtapi2 Logs)

Neuer Anruf und Umleitung an verfügbaren Port



Neuer Anruf und Umleitung an verfügbaren Port

```
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
}

```

Neuer Anruf an CTI-Port

```

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
}

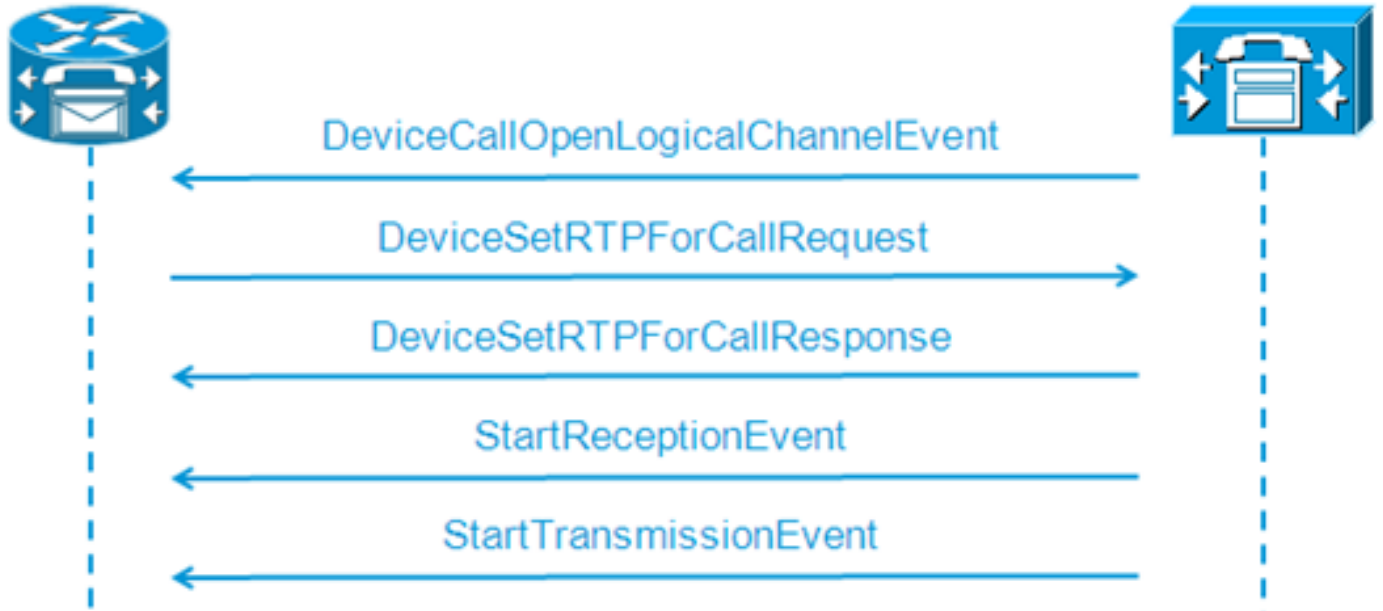
```

```
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-Port akzeptiert umgeleiteten Anruf

```
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
```

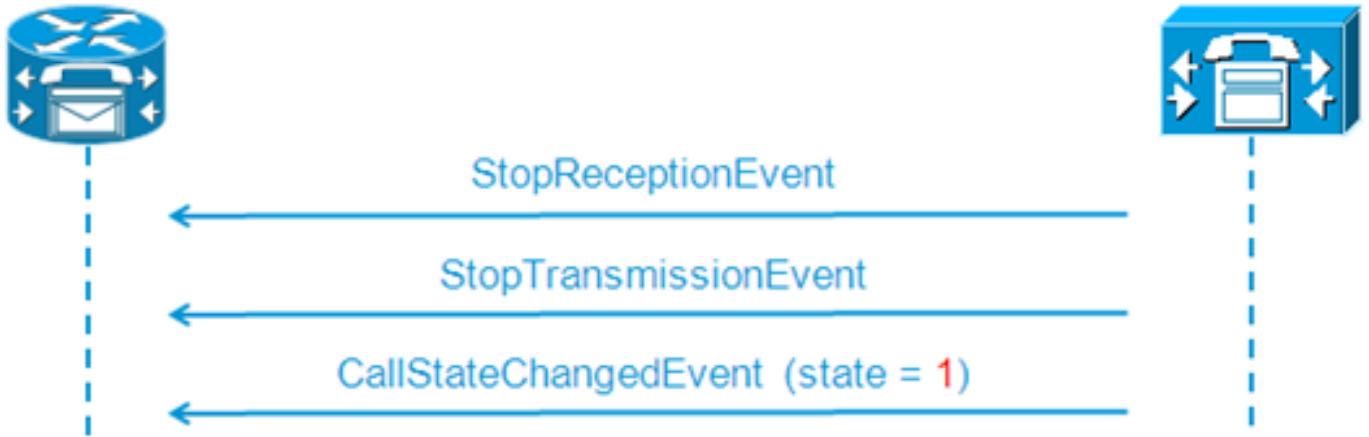
Medienverhandlungen



```

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
}
  
```

Anruftrennung



```

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-Ein/Aus-Signalisierung

CUE schaltet MWI-Lampe für Leitung 3001 ein

```

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
}

```

Gewählte DTMF-Nummer '3' zum Löschen der Nachricht aus dem Postfach

```

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 schaltet MWI-Lampe für Leitung 3001 aus

```
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
}
```

Echtzeit-CCN-Protokolle

(Ergebnisse aus den Echtzeit-CCN-Protokollen)

So wird derselbe Anruf aus dem vorherigen Beispiel hier angezeigt, wenn stattdessen die Echtzeit-CCN-Protokolle gesammelt werden.

Anrufeinrichtung

```
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
```

Anruftrennung

```
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
```

Problembhebung - Anwenderberichte

Verbindungsprobleme

In diesem Szenario sind die CUE-Ports und -Trigger aufgrund der fehlenden Verbindung zwischen CUE und CUCM nicht beim CUCM registriert.

```
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 --

```

Hinweis: Die Zeitstempelsekunden liegen zwischen 13:20:28 und 13:20:38; Aus diesem Grund können wir feststellen, dass CUE den TCP-Socket 10 Sekunden lang nicht öffnen konnte, bevor die Unfähigkeit erkannt wurde, einen Anbieter zu erstellen.

Authentifizierungsprobleme

In diesem Szenario können sich die CUE-Ports und -Trigger nicht beim CUCM registrieren, da die zwischen CUE und CUCM konfigurierten Passwörter nicht übereinstimmen.

CCN-Protokoll

```

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

```

Benutzer nicht CTI-fähig

In diesem Szenario können sich die CUE-Ports und -Trigger nicht beim CUCM registrieren, da der JTAPI-Anwendungsbenutzer der Berechtigungsgruppe **Standard CTI Enabled** nicht auf der CUCM-Seite hinzugefügt wurde. Selbst wenn sich die Benutzeranmeldeinformationen dementsprechend authentifizieren, kann der JTAPI-Benutzer, in diesem Fall **tacjtapiuser**, keine Geräte über CTI und JTAPI steuern.

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
qbcClientVersion = 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
}

```

Der CUCM CTI Manager-Dienst ist ausgefallen.

In diesem Szenario können sich die CUE-Ports und -Trigger nicht registrieren, da der CUCM CTI Manager-Dienst ausgefallen ist oder sich in einem ungewöhnlichen Status befindet. Es erhält einen Fehler "Verbindung verweigert" für den Verbindungsversuch des CUE mit dem JTAPI-TCP-Port 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
```

Konfigurationsfehler

In diesem Szenario kann CUE den JTAPI-Trigger nicht mit der Nummer 9999 registrieren, da der abzugleichende CTI-RP nicht konfiguriert ist oder den "steuerbaren Geräten" für den Benutzer auf der CUCM-Seite nicht hinzugefügt wurde. Der CUE stellt dies fest, nachdem er die **GetDeviceInfoFetchResponse** vom CUCM erhält und bemerkt, dass sich in der Anbieterdomäne kein Gerät befindet, das auf alle vom Benutzer steuerbaren Geräte verweist, die mit der Trigger-Nummer übereinstimmen, die er lokal konfiguriert hat. Der CUE versucht dann nicht, eine **DeviceOpenRequest** für diesen bestimmten Trigger zu senden, sondern meldet stattdessen nur die Ausnahme in den Ablaufverfolgungen. Das CUE versucht weiterhin, alle anderen Geräte in der Domäne des Anbieters zu registrieren, die vom CUCM gesendet werden.

```
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.

```

Hinweis: Auch wenn der Trigger 9999 lokal in CUE konfiguriert ist, ist er nicht Teil der Provider-Domäne, die vom CUCM empfangen wurde, und registriert sich daher nicht.

CUE öffnet weiterhin Zeile 8000; die in der Domäne des Anbieters enthalten ist.

```

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-Anrufweiterleitungsproblem

In diesem Szenario ruft der Benutzer mit DN 3001 das CUE-Modul auf, um sein virtuelles System zu überprüfen. Der Anruf wird dem VM-Pilot (CTI RP) von CUE mit DN 8000 angezeigt. Das CUE fordert dann den Anruf an, an seinen CTI-Medienport mit DN 8501 umgeleitet zu werden. Der Anruf wird jedoch nicht umgeleitet, da der für DN 3001 konfigurierte CSS keinen Zugriff auf den PT hat, dem die CTI-Port-DN zugewiesen ist.

```

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

```

Lizenzprobleme

In diesem Szenario kann CUE seine Ports und Trigger nicht registrieren, da die Lizenzen für die VM-Ports nicht aktiviert wurden. Aus demselben Grund werden in den Captures keine Registrierungsversuche angezeigt.

Zusammenfassung aus der dekodierten atrace.log:

```

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

```

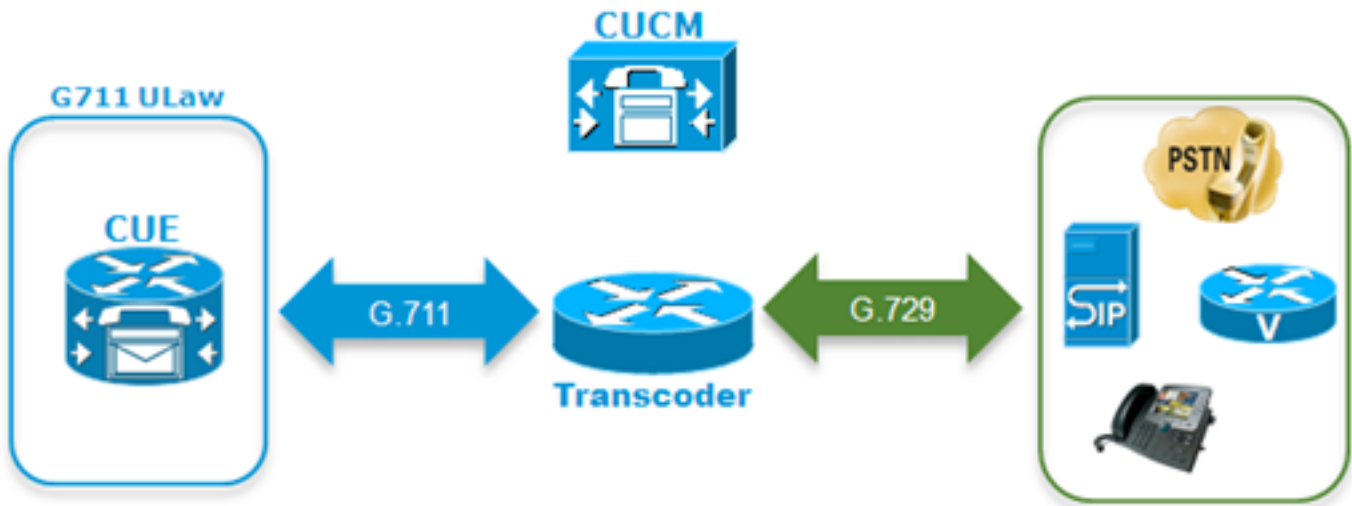
activation count has been set to zero

```
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
```

Best Practices

CUE unterstützt nur den G711ulaw-Codec. Daher ist in fast jeder Bereitstellung ein Transcoder erforderlich, damit CUE mit anderen Geräten oder Trunks kommunizieren kann, die verschiedene Codecs verwenden (einschließlich G711Alaw). Dasselbe gilt für DTMF-Interworking mit Geräten, die nur In-Band-DTMF unterstützen, bei denen auch eine MTP-Ressource (Media Termination Point) erforderlich ist. Aus diesen Gründen empfiehlt Cisco Folgendes:

- Erstellen Sie einen isolierten Gerätepool, um nur die CTI-RPs und CTI-Ports von CUE zu verwenden. Falls mehr als ein CUE in CUCM integriert ist, erstellen Sie einen Gerätepool pro CUE.
- Erstellen Sie einen separaten Bereich nur für den RP und die Ports des CUE-Routers, und wenden Sie ihn auf diesen isolierten Gerätepool an.
- Stellen Sie sicher, dass die Region so konfiguriert ist, dass nur G711 mit allen anderen Regionen zugelassen wird.
- Stellen Sie sicher, dass eine MRGL (Media Resource Group List) mit verfügbaren Umkodierungsressourcen auf den Gerätepool der CTI-RP- und CTI-Ports von CUE angewendet wird, sodass diese bei Bedarf Zugriff auf eine Transcoder-Ressource haben.



- Wenn der Benutzer nicht durch die Sprachmenüs mit DTMF-Tönen navigieren kann, muss der MRGL der CUE-Geräte möglicherweise eine MTP-Ressource hinzugefügt werden.

Erstellen eines separaten VM-Profiles für CUE im CUCM

Um in letzter Zeit auftretende Probleme mit dem CUCM-CTI-Manager zu vermeiden, wird empfohlen, alle Telefone dem CUE-JTAPI-Benutzer auf CUCM-Seite zuzuordnen und nicht nur den CTI-RPs und -Ports.

Wenn die SRST-Funktion (Survivable Remote Site Telephony) gewünscht wird:

- Stellen Sie sicher, dass für jeden JTAPI-Trigger auf dem CUE-Modul der entsprechende SIP-Trigger konfiguriert ist.
- Stellen Sie sicher, dass dem Zweigstellen-Router DFÜ-Peers hinzugefügt werden, damit Anrufe im SRST-Modus über SIP an das CUE-Modul weitergeleitet werden können.
- Konfigurieren Sie die Maske für externe Nummern der CTI-Weiterleitungspunkte sowie die Maske für das CFU-Feld (Call Forward Unregistered) im CUCM, um sicherzustellen, dass der CUCM die für das Zweigstellenmodul vorgesehenen Anrufe über das PSTN-Gateway weiterleitet, wenn die Verbindung zwischen CUCM und CUE verloren gegangen ist oder automatisiertes Alternativrouting (AAR) aufgerufen wird. Möglicherweise sind zusätzliche Übersetzungsregeln erforderlich, damit der Zweigstellen-Router eingehende Anrufe vom PSTN an das CUE-Modul weiterleiten kann.
- Wenn der Konfigurationsansatz Direct Transfer to VM auf der CUCM-Seite vorhanden ist und der Benutzer diese Funktionalität bei CME-SRST beibehalten möchte, müssen Sie die alte Dummy-DN mit dem CFA-Konfigurationsansatz verwenden, der für CME verwendet wurde, bevor der **Softkey TransferToVM** verfügbar wurde. Weitere Informationen finden Sie unter [Weiterleiten eines Anrufers direkt an ein Unity Express-Postfach](#). Hier ein Beispiel, wie das aussehen kann. Beachten Sie, dass dies nur möglich ist, wenn **CME-SRST** verwendet wird, nicht aber mit Legacy-SRST mit Fallback von Anrufmanagern. Angenommen, die DNs liegen im Bereich von 200 bis 299. Der Anruf kommt für x201 ein. Konfigurieren Sie eine **ephone-dn** mit dem folgenden Befehl:

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

Zeigen Sie im DFÜ-Peer auf CUE: Verwenden Sie eine Regel und ein Profil für die ausgehende Übersetzung, um das vorangestellte Sternchen ("*") zu entfernen und den **Redirect Dialed Number Information Service (RDNIS)** wieder auf die ursprüngliche dreistellige Nummer (z. B. 201) oder die vollständige E.164-Nummer zu ersetzen, falls die **Telefonnummer** mit der vollen DID in der CUE konfiguriert wurde. Stellen Sie sicher, dass der Diversion-Header der an CUE gesendeten INVITE-Nachricht mit der für den Benutzer auf der

CUE-Seite konfigurierten **Telefonnummer** übereinstimmt.

Checkliste für die Portregistrierung Fehlerbehebung

1. **Überprüfen Sie die Konfiguration auf CUCM-Seite:** Sind die Webdienste CTI Manager, CallManager und Administrative XML (AXL) aktiviert und gestartet? Wurden die CTI-Ports und Routing-Points konfiguriert und einem eindeutigen DN zugewiesen? Ist die JTAPI-Benutzer-CTI aktiviert und verfügt sie über Zugriff auf die AXL API? Verfügt der JTAPI-Benutzer über die Kontrolle aller CTI-Routing-Punkte und -Ports? Manchmal empfiehlt es sich, den CTI Manager-Dienst nach dem Hinzufügen der Konfiguration auf allen Servern neu zu starten. Dies kann sich jedoch negativ auf den Service auswirken und empfiehlt sich, ein Wartungsfenster zu planen, da dies alle anderen Geräte betrifft, die CTI und JTAPI mit dem CUCM verwenden, z. B. Unified Contact Center Express (UCCX), IP Manager Assistant (IPMA), Attendant Console, AA- oder ACD-Anwendungen von Drittanbietern usw.
2. **Überprüfen Sie die Konfiguration auf der CUE-Seite:** Ist der **Call-Agent** als CUCM definiert? Wurden die Port-Lizenzen aktiviert? Evaluierungslizenzen sind für die Erstkonfiguration zulässig. Können Sie einen Ping an den CUCM senden? Wurden die Anmeldeinformationen für den JTAPI-Benutzer korrekt hinzugefügt, und wurden die **Call Agents** definiert? Wurde das Modul neu geladen, sodass die Konfigurationsänderungen übernommen wurden? Wenn der CTI-RP und der Port nicht automatisch aus dem CUCM importiert werden, versuchen Sie, die Port-DNs unter **ccn subsystem jtapi** manuell hinzuzufügen, sowie die **Jtapi**-Trigger für jeden CTI-RP und laden Sie das Modul neu.

Wenn alle diese Elemente bestätigt werden, erhalten Sie im nächsten Schritt JTAPI-Ablaufverfolgungen für CUE und möglicherweise CUCM CTI-Ablaufverfolgungen, um das Problem weiter zu isolieren.

Zugehörige Informationen

- [Einrichtung und Erfassung von Trace-Daten in CUE](#)
- [Fehlerbehebung bei Cisco Unity Express-Systemfunktionen](#)
- [Konfigurationsbeispiel für CallManager für Cisco Unity Express](#)
- [Konfiguration des CUCM-Voicemail-Profiles](#)
- [Bereitstellungsmodelle für Cisco Unity Express](#)
- [Best Practices für die Bereitstellung von Cisco Unity Express](#)
- [CUCM-Transkodierung](#)
- [CUCM-MTP](#)
- [Technischer Support und Dokumentation - Cisco Systems](#)