

JTAPI-architectuur en gespreksstroom in UCX begrijpen

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Achtergrondinformatie](#)

[JTAPI - Overzicht](#)

[JTAPI en UCCX](#)

[JTAPI-architectuur](#)

[JTAPI Observer-model](#)

[JTAPI-provider-model](#)

[JTAPI-gebruiker](#)

[P1- en P2-handgrepen](#)

[Call Flow](#)

[Problemen oplossen](#)

[JTAPI-logbestanden inschakelen en verzamelen](#)

[JTAPI-debuggs inschakelen](#)

[JTAPI-debuggs verzamelen](#)

[Van RTMT](#)

[Van CLI](#)

[JTAPI-loglocatie](#)

[Referentie links](#)

Inleiding

Dit document beschrijft de basisfunctionaliteit van JTAPI, de architectuur en de gespreksstroom met betrekking tot UCCX.

Voorwaarden

Vereisten

Cisco raadt kennis van deze tools en functies aan:

- JTAPI - API voor Java-telefonie
- API - Toepassingsprogrammeerinterface

- UCS - Unified contactcenters Express
- CUM - Cisco Unified Communications Manager
- CTI - Computer Telephony Integration

Cisco raadt kennis van deze onderwerpen aan:

- Configuratie Cisco Unified Communications Manager (CUCM)
- Configuratie van Unified Contact Center Express (UCX)

Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende softwareversies:

- UCS UCS versie 12.5.1.11002-481
- CUCM-versie 12.5.1.11900-146

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Achtergrondinformatie

Dit document beschrijft de architectuur van JTAPI, de vraagstroom, het toelaten van debugs en het verzamelen van de logboeken.

JTAPI - Overzicht

Cisco Unified JTAPI fungeert als een programmeerinterfacestandaard die door Sun Microsystems is ontwikkeld voor gebruik met op Java gebaseerde toepassingen voor computertelefonie. Cisco JTAPI implementeert de Sun JTAPI 1.2 specificatie met extra Cisco-extensies. U moet Cisco JTAPI gebruiken om toepassingen te ontwikkelen die:

- Besturing en waarneming van Cisco Unified Communications Manager-telefoons.
- Routeaanroepen met behulp van Computer-Telephony Integration (CTI) poorten en routepunten (virtuele apparaten).

JTAPI en UCCX

Cisco Unified JTAPI wordt in een contactcenters gebruikt om de status van het apparaat te bewaken en om routinginstructies uit te geven om gesprekken op de juiste tijd naar de juiste plaats te verzenden. Naast het, wordt JTAPI gebruikt om opnameinstructies te beginnen en tegen te houden terwijl het terugwinnen van vraagstatistieken voor analyse en aan scherm-pop vraag in CRM toepassingen, geautomatiseerde scripting, en afstandsbediening van de vraag.

JTAPI-architectuur

Cisco Unified JTAPI, gebruikt in een ondernemingsomgeving, combineert beschikbaarheid, locatie en voorkeuren van gebruikers voor een unieke omgeving op maat voor op aanwezigheid gebaseerde routing.

Hier zijn de toepassingen van JTAPI:

- JTAPI kan twee of meer telefoon en lijncombinaties controleren of op de hoogte worden gesteld.
- De Toepassingen van het contactcentrum gebruiken Volledig Gespreksmodel voor JTAPI.
- JTAPI biedt deze functionaliteit:
 - Gespreksbeheer
 - Mediacontrole
 - Mediaonderhandeling

JTAPI Observer-model

Het volgende diagram verklaart het Provider-Observer model dat JTAPI werkt.

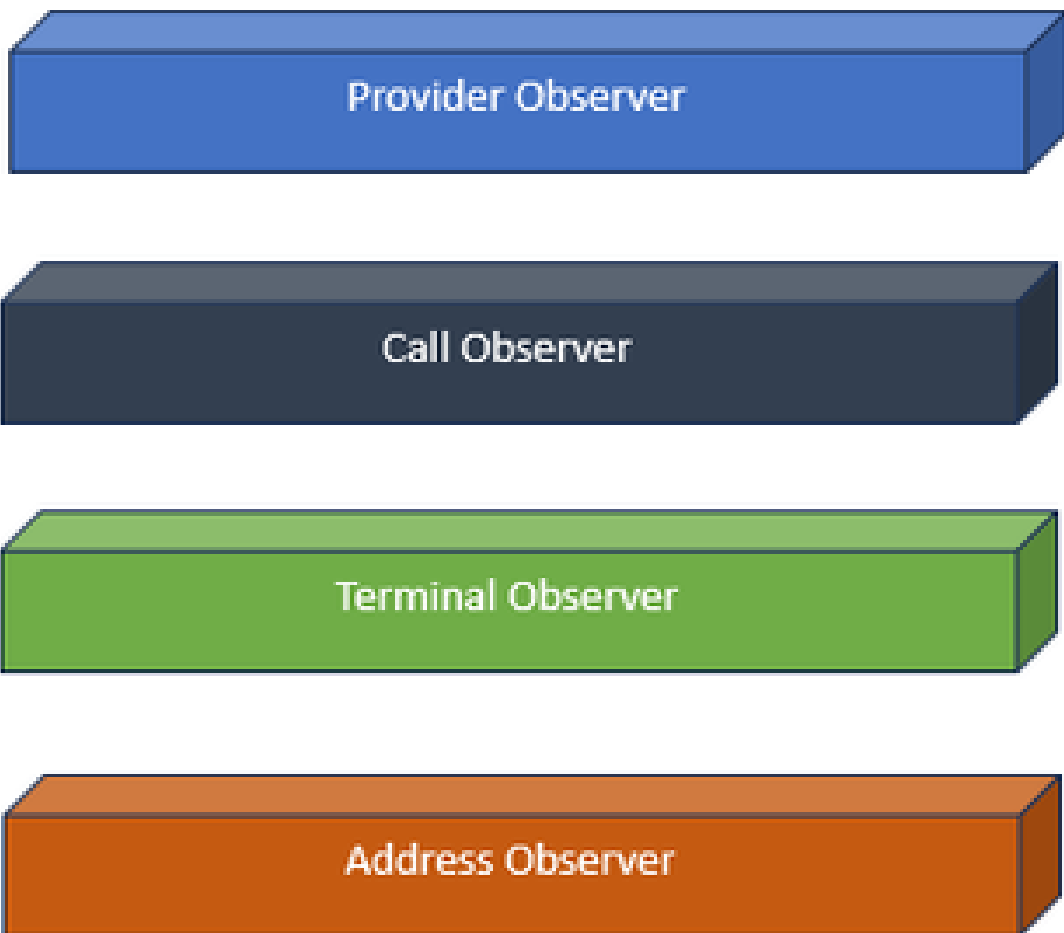
Provider

Call

Terminal

Address

Waarnemers



Interfaces voor waarnemers

JTAPI is gebaseerd op het idee van waarnemer. Door waarnemer verwijst het naar het idee van degene die de gebeurtenissen waarneemt. Je kunt meerdere waarnemers plaatsen op verschillende dingen binnen het systeem. JTAPI gebruikt waarnemers om te weten te komen over de staatswijzigingen van de objecten.

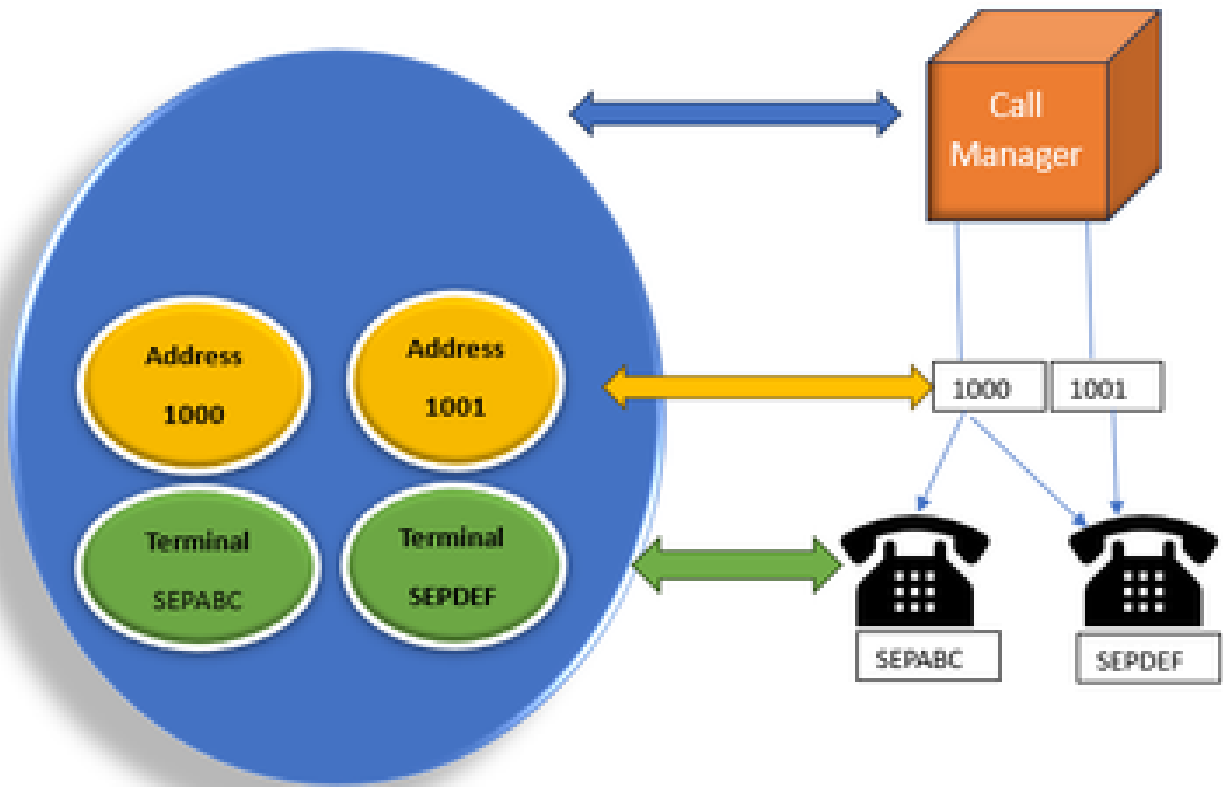
Je kan bijvoorbeeld waarnemers plaatsen op lijnen, telefoons, terminal, adressen enzovoort, en leren over hun status veranderingen.



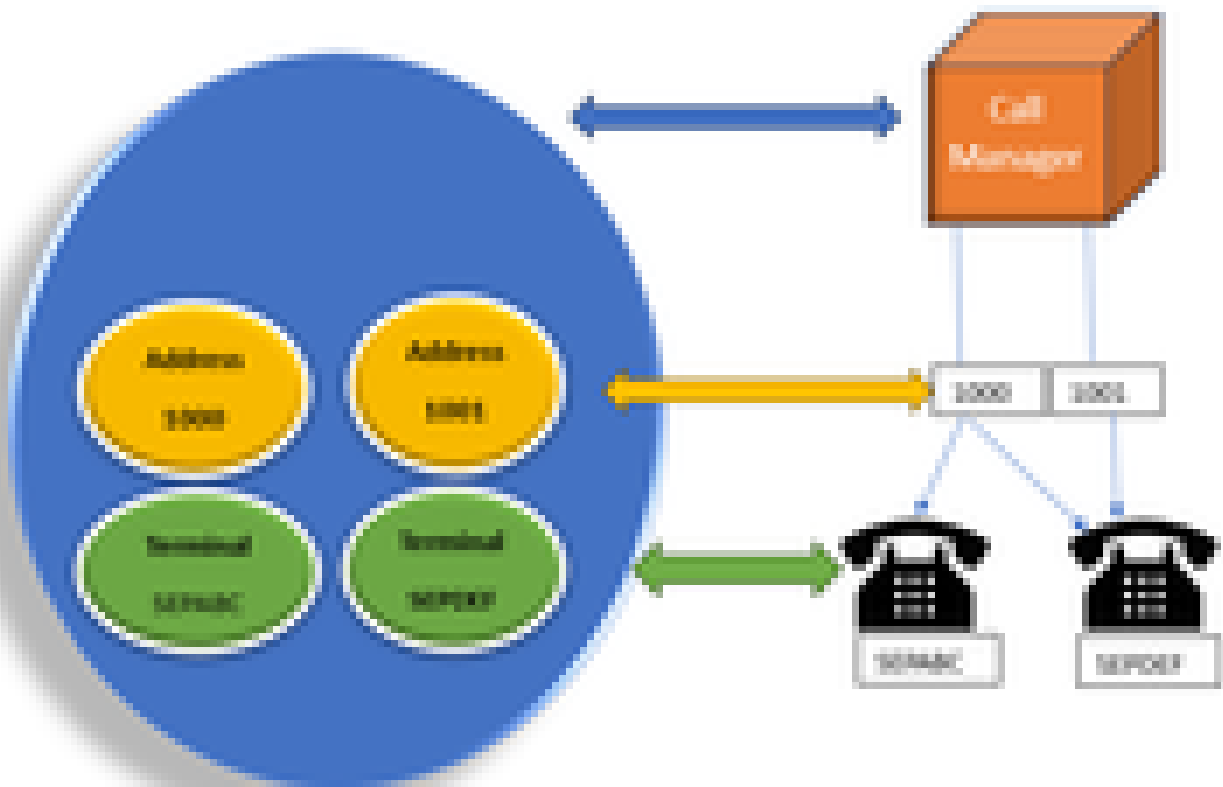
Opmerking: als er geen waarnemers op een object zijn geplaatst, kunt u niet op de hoogte worden gesteld van de maatregelen die op hen zijn genomen.

JTAPI-provider-model

Het volgende diagram verklaart het model van de Leverancier dat JTAPI werkt:



JTAPI-

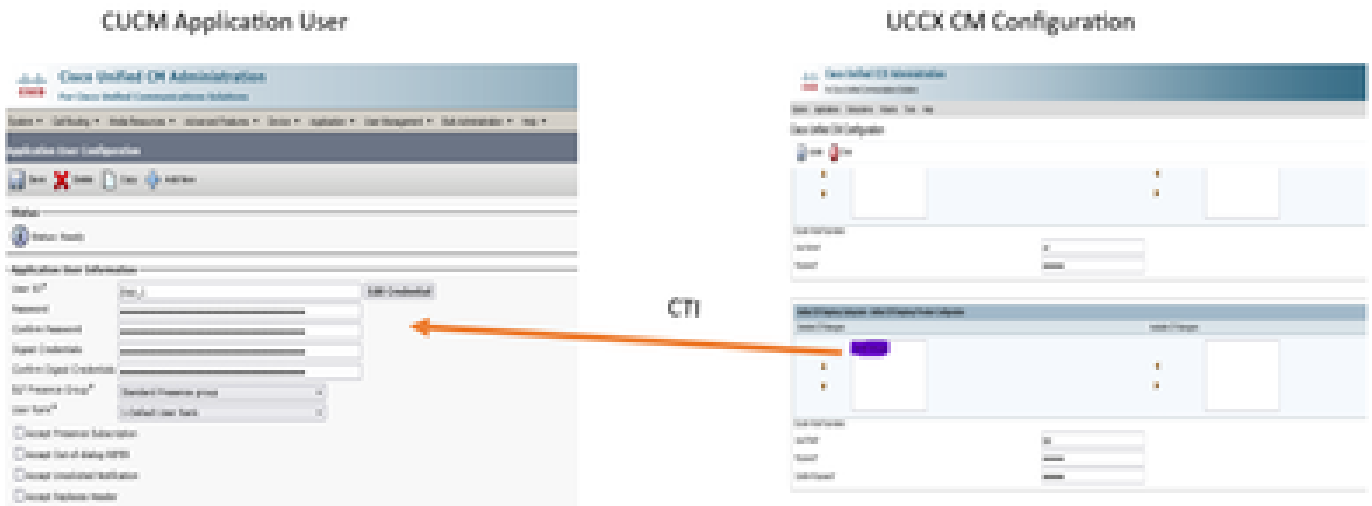


provider-model

JTAPI-provider is een verbinding van de toepassing naar Call Manager of CTI Manager. De leveranciers worden gebruikt om waarnemers aan de voorwerpen vast te maken. U kunt ook een waarnemer aan een provider toevoegen. Aanbieders worden gebruikt om op de hoogte te worden gesteld van de maatregelen die worden genomen met betrekking tot de objecten. (U kunt ook de controle overnemen over de apparatuur waarop de waarnemer is aangesloten (van de aanbieder die deze waarnemer heeft aangesloten).

JTAPI-gebruiker

De volgende screenshots worden genomen van de CUCM (links) en UCCX (rechts) die uitleg geven over de JTAPI-toepassingsgebruiker.



- Wanneer u een toepassing start en een verbinding wilt maken met de CTI-beheerder, opent u een provider.
- De reden om een provider te openen is dat u de acties die worden uitgevoerd op de apparaten, terminals, enzovoort kunt controleren.
- De manier waarop het wordt geïmplementeerd in CUCM is via een Application-gebruiker.
- U maakt deze gebruiker en geeft referenties zodat het eerst via CTI naar de CUCM kan worden geverifieerd.
- De JTAPI-toepassingsgebruiker die in de CUCM is gemaakt, is dus de provider.
- Naast alleen controle, moet u ervoor zorgen dat die apparaten onder **Bijbehorende Apparaten** zijn om ervoor te zorgen dat u de apparaten kunt controleren.



Opmerking: u maakt de JTAPI-gebruiker niet op cucm, dit wordt gemaakt door de toepassing (UCCX) via AXL op CUCM.

P1- en P2-handgrepen

P1 en P2 zijn de Provider Handgrepen. Deze worden belangrijk wanneer u meerdere providers van dezelfde applicatie gaat openen. Vanuit UCCX maakt u 2 providers, waarvan één de controle heeft over de CTI-poorten en routepunten, de andere de controle heeft over de agent-telefoons die RM-JTAPI provider worden genoemd. U, als UCCX creëert de provider die de CTI poorten en routepunten eerst bestuurt, dat is de JTAPI P1 provider. De andere provider die u na de P1 maakt, is de P2-provider of de RmCm-provider die de agent-apparaten verwerkt.

Als u P1 nieuwe vraaggebeurtenis ziet, weet u dat een vraaggebeurtenis van een haven CTI of een routepunt CTI is. Als u P2 nieuwe vraaggebeurtenis ziet, weet u dat is een vraag gebeurtenis van de daadwerkelijke telefoon. U maakt één RM-JTAPI-gebruiker en één JTAPI-gebruiker aan de CCX-kant, maar aan de CUCM-kant zie je 2 JTAPI-gebruikers gemaakt. Dit komt doordat elke CCX-knooppunt zijn eigen JTAPI-

gebruiker krijgt, maar ze delen de één RM-JTAPI-gebruiker. Wanneer u een Trigger op UCCX maakt, wordt deze toegevoegd aan beide JTAPI-gebruikers. Beide knooppunten openen afzonderlijk een provider. Dus elke actie die ondernomen wordt op routepunt wordt op beide CCX-knooppunten gemeld.

Voor het overige, het secundaire knooppunt zit daar gewoon en blijft erop wijzen dat het nog steeds het secundaire knooppunt is. Elke knooppunt heeft een afzonderlijke groep CTI-poorten. De CTI-poorten van knooppunt 1 worden bestuurd door de jtapi_1. De CTI-poorten van knooppunt 2 worden bestuurd door jtapi_2.

Wanneer de oproep op het CTI-routepunt aankomt, worden beide CCX-knooppunten op de hoogte gesteld van de nieuwe gespreksgebeurtenis, maar de actieve CCX-knooppunt reageert op de Call Manager met een verzoek om een van de CTI-poorten die hij controleert opnieuw te sturen.

Als u een IP tegen een CTI-routepunt in CUCM ziet, is het een van de IP-adressen van de CCX, maar dat betekent niet dat bepaalde knooppunten de oproep routeren.

Een gemeenschappelijk ding dat je doet is dat we het apparaat loskoppelen van de JTAPI-gebruiker en het opnieuw toevoegen. De reden achter het is wanneer u het loskoppelt, wordt de provider op de hoogte gebracht over het en verwijdert alle verbindingen naar het en wanneer het opnieuw wordt toegevoegd, wordt de waarnemer opnieuw toegevoegd en begint de provider het opnieuw te controleren met behulp van open provider verzoek. Je kunt zien dat behalve het apparaat dat wordt toegevoegd in de lijst van gecontroleerde apparaten, er is een ander ding genaamd **Allow Control of Device from CTI** checkbox op het individuele apparaat ook. Dit moet leiden tot een extra niveau van granulariteit. Dus als u het routepunt toegevoegd in de lijst van gecontroleerde punten maar CTI checkbox is niet aangevinkt, kunt u nog steeds een bericht krijgen over de gebeurtenissen op dat routepunt, maar er zijn geen acties op call control mogelijk.

Call Flow

Hier zijn de sequenties van gebeurtenissen die betrokken zijn bij de UCCX-gespreksstroom:

- Wanneer de oproep bij het CTI-routepunt aankomt, wordt deze doorgestuurd naar een cti-poort.
- CTI-poort opent het mediakanaal met het ipvms-stuurprogramma in de uccx-box.
- Zodra u besluit dat de agent de oproep moet nemen, wordt een consult-overdracht gedaan van de poort naar de agent.
- De nieuwe vraaggebeurtenis wordt verzonden naar het Punt van de Route CTI.
- Aanvraag voor omleiding gaat naar CTI-poort.
- De staat van de call-id wordt niet ingevuld.
- Dan komt er een nieuwe call event voor de oproep naar de CTI-poort.
- Als de redirectierespons 0 is, is het goed, als niet nul, betekent het ontbroken.
- Dan stuurt u vraag accepteert verzoek (dit wordt niet beantwoord, net geaccepteerd op poort).
- Dan accepteer stap hits op het script, dit is wanneer vraag antwoord verzoek komt in Jtapi.



Opmerking: dit gebeurt zo snel en soms zijn er meerdere gebeurtenissen op hetzelfde moment, zoals oproepen die komen van Cisco Unity Connection of overdracht die tijd van CUCM nemen, dit kan race-conditie in de acceptatiestap veroorzaken, wat de reden is waarom u dit wilt vertragen. Dit doet u door vertragingstap toe te voegen voordat u stap accepteert.

11. Dan krijg je een antwoord van de haven.

12. De gespreksstatus verandert in verbonden.



Opmerking: CTI is asynchroon in tegenstelling tot sip of skinny telefoons die wachten op de reactie voordat een nieuw verzoek wordt verzonden, wat de reden is waarom de volgorde van berichten in de JTAPI CTI-berichten kan worden verward.

13. Nadat de haven antwoord heeft gegeven, wordt er onderhandeld over de media.

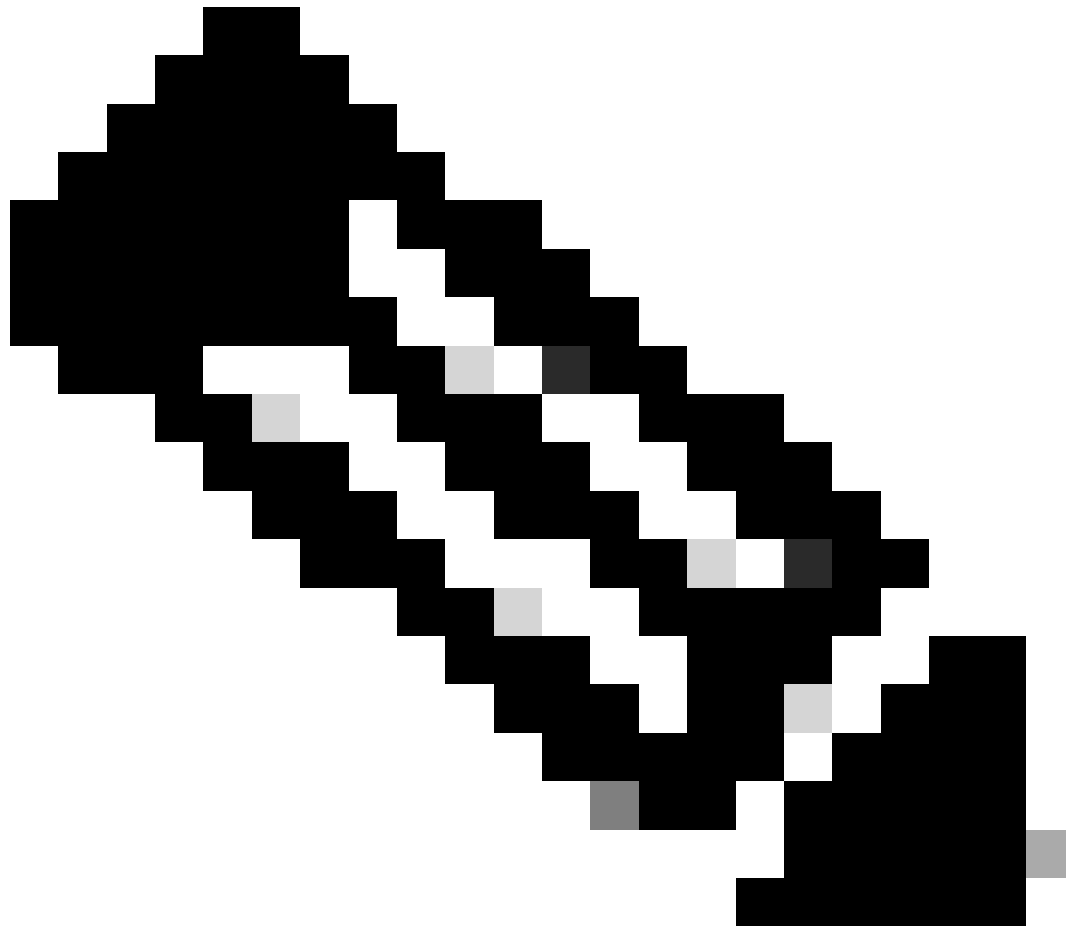
14. Port verstuurt een open_logical_channel-verzoek waarin het zijn poort en ip deelt waarop het de externe partij de RTP naar wil sturen.

15. Alvorens het logische kanaal te openen, maakt het een verbinding met het IPVMS-stuurprogramma aan op het UCCX-vak en opent het een RTP-stream.

16. Volgende gebeurtenis is de start_receptie gebeurtenis die ons de verre eindinformatie vertelt (d.w.z. de ip en de poort van het oproepende apparaat).

17. Dan krijg je start_transmissie event net als elk ander mediasignaal.

18. Nu luistert u naar de IVR en de aanwijzingen.



Opmerking: dit is waar de installatie van de oproep voltooid is.

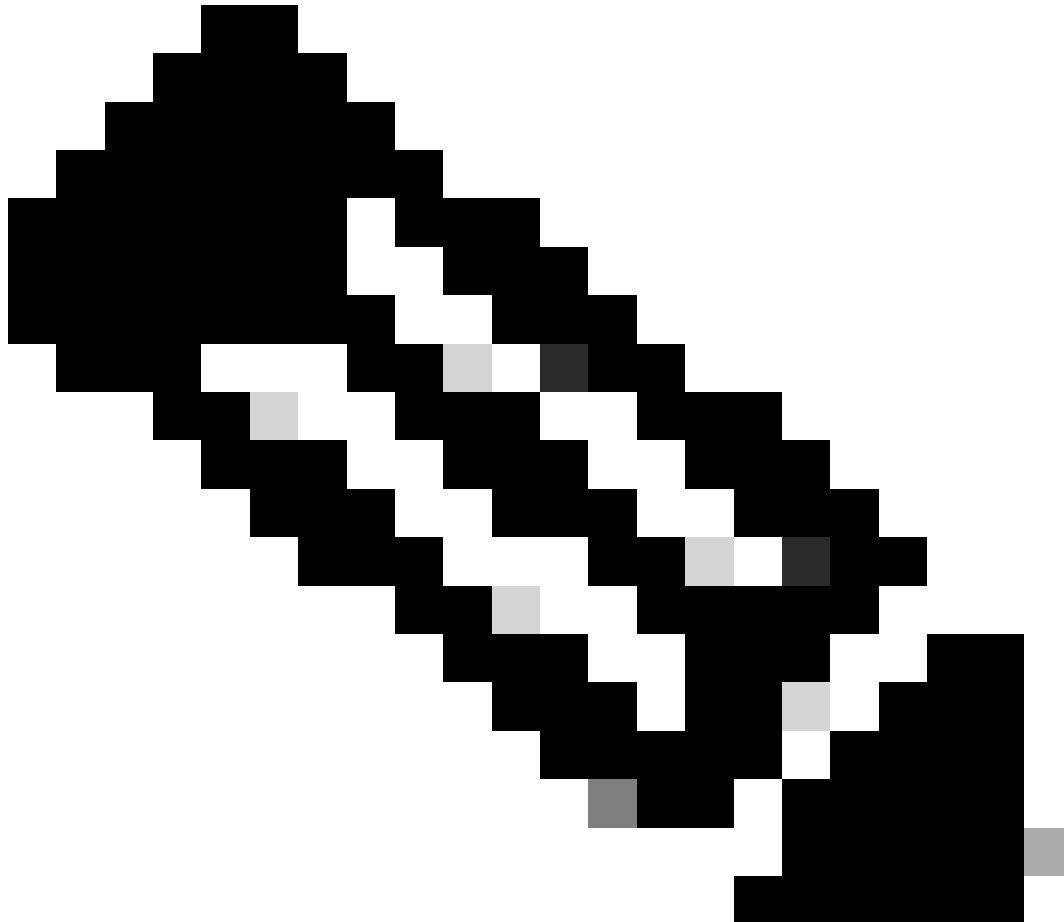
19. Nu wanneer de oproep een stap in het script bereikt die het mogelijk maakt de oproep naar de agent over te brengen, ziet u een CallSetupTransferrequest.

20. In tegenstelling tot het eerste bericht is dit een consultatie-overdracht en geen doorverwijzing.

21. De reden waarom dit een consultatie-overdracht is, is omdat als een agent KLAAR is, maar niet op zijn plaats, en we de oproep doorsturen, het gewoon onbeantwoord blijft, maar als we een consultatie-overdracht doen, dan als de agent er niet is, dan wordt de oproep opnieuw geplaatst

in de wachtrij.

22. Net als elke andere consulent-overdracht, is dit de CTI-poort die de transferknop voor het eerst op de telefoon raakt.



Opmerking: ConsultWithoutMedia is de API voor de consult-overdracht.

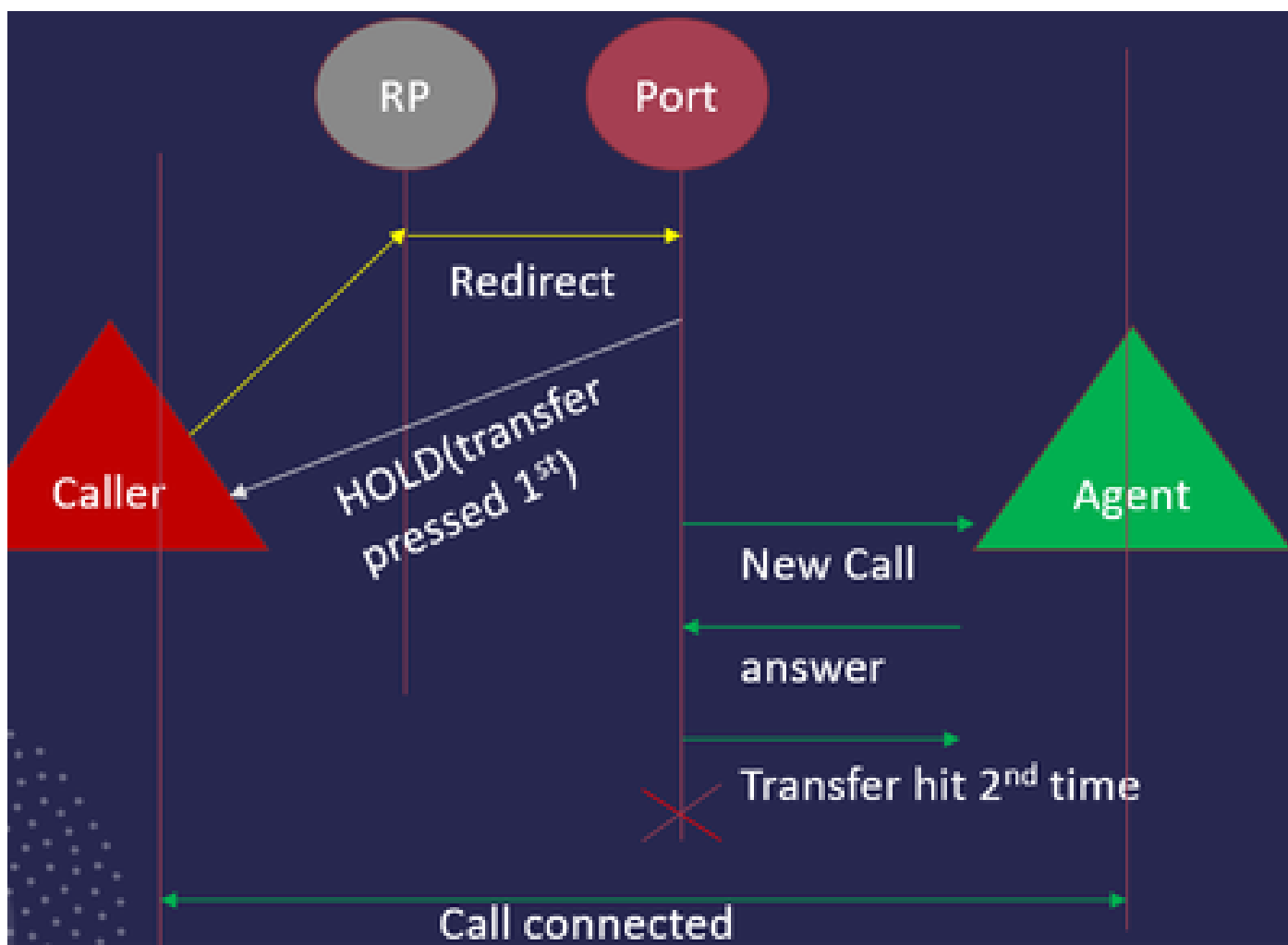
23. In een reguliere consultatie-overdracht is er een mediapad tussen A en C, maar in dit gelast u CUCM dit niet te doen, omdat het geen zin heeft dat u media tussen de UCCX en de Agent maakt.

24. Dus u gaat een consult-overdracht doen zonder een media-cut-over te doen tussen de agent en de CTI-poort.

25. Op dit punt, zet de CTI haven de bezoeker op greep en wij zien een vraagstaat veranderde event=HOLD.

26. Nu ziet u een nieuwe call-gebeurtenis van de CTI-poort naar het agent-apparaat (met behulp van de oorspronkelijke call-id, maar een subset daarvan en een P2-gebeurtenis).
27. Als u de P2-eventoproepid doorzoekt, ziet u het volgende bericht als CallAntwoordverzoek wat betekent dat de agent de oproep heeft opgepikt.
28. Zodra u weet dat de agent de oproep heeft ontvangen (met behulp van P2) en de oproep ook in verbonden staat aan de CTI-poortkant is (met behulp van P1), ziet u een routepunt CallDirectTransferRequest, wat betekent dat de overdrachtknop voor de tweede keer is ingedrukt.
29. Nu de CTI-poort weet dat de agent de oproep heeft opgepikt, is er geen reden om te wachten, wordt er onmiddellijk een CallDirectTransferRequest verzonden dat de CTI-poort is die de overdrachtknop tweede keer indrukt, zoals hierboven is uitgelegd.
30. Nu is het oorspronkelijke oproepbeen (dat in de wachtstand stond) het overlevende.
31. De andere telefoonpoot wordt vernietigd (tussen haven en agent).
32. Op dit punt, CUCM en UCCX uit het beeld en RTP wordt gevestigd tussen de bezoeker en de agent door Gateway.

Het volgende diagram verklaart de eerder vermelde vraagstroom op een samengevatte manier.



JTAPI Call Flow Samenvatting

JTAPI-logbestanden inschakelen en verzamelen

JTAPI-debug inschakelen

Controleer deze stappen om JTAPI debug levels in te schakelen.

- Log in op UCCX.
- Ga naar **CCX-onderhoudsnet**.
- Ga naar **Trace**.
- Ga naar **Configuratie**.
- Selecteer in de vervolgkeuzelijst Select Service de optie **Cisco Unified CM Telephony Client**.
- Selecteer het **aanvinkvakje Debugging**.
- Selecteer alle selectievakjes behalve **MISC_DEBUGGING**.

JTAPI-debug verzamelen

Controleer deze stappen om JTAPI debug niveaus van RTMT of CLI in te schakelen.

Van RTMT

- Meld u aan bij CCX Real Time Monitoring Tool.
- Klik op **Trace & Log Central**.
- Klik op **Collect Files**.
- Selecteer **JTAPI-client** voor alle servers.
- Klik op **Next** (Volgende).
- Selecteer de tijdstempels en de locatie waar u de logbestanden wilt opslaan.

Van CLI

JTAPI-loglocatie

activelog/uccx/log/JTAPI

Opdracht om de JTAPI-logbestanden te verzamelen:


```
file get activelog/uccx/log/JTAPI/* recurs compress
```

Syntaxis:

```
file get {activelog|inactivelog|install} file-spec [Options]
```

file-spec verplicht bestand om over te dragen

opties optionele reltime maanden|weken|dagen|uren|minuten tijdswaarde

abstime hh:mm:MM/DD/JJU:mm:MM/DD/JJ

match regex

terugkomen

samenpersen

5 manieren om de logbestanden te downloaden op basis van de tijdstempel

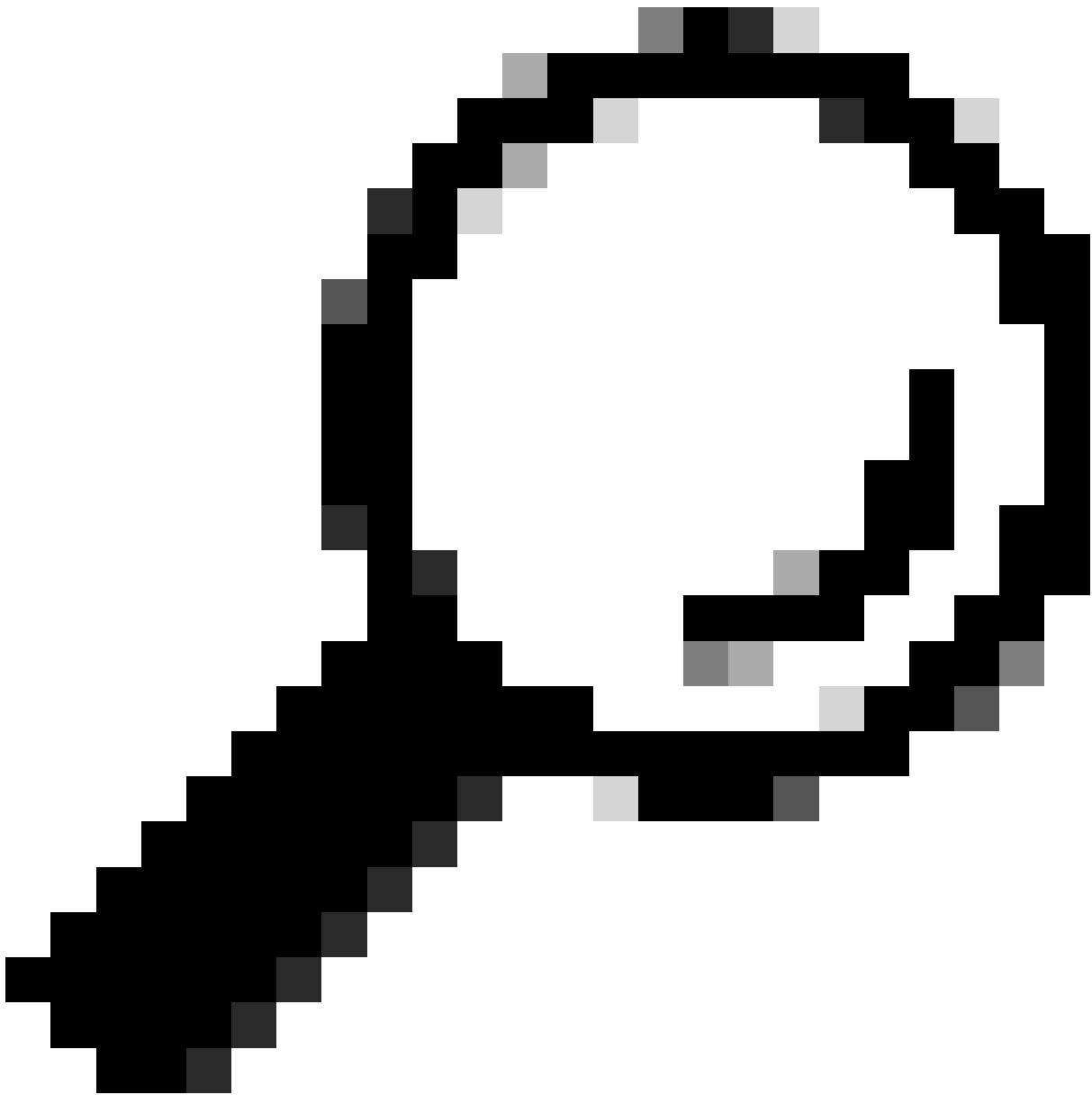
reltime—Relatieve tijdsperiode, opgegeven als minuten | uren | dagen | weken | maandenwaarde

abstime—Absolute tijdsperiode, zoals aangegeven in hh:mm:MM/DD/JJH:mm:MM/DD/JJ

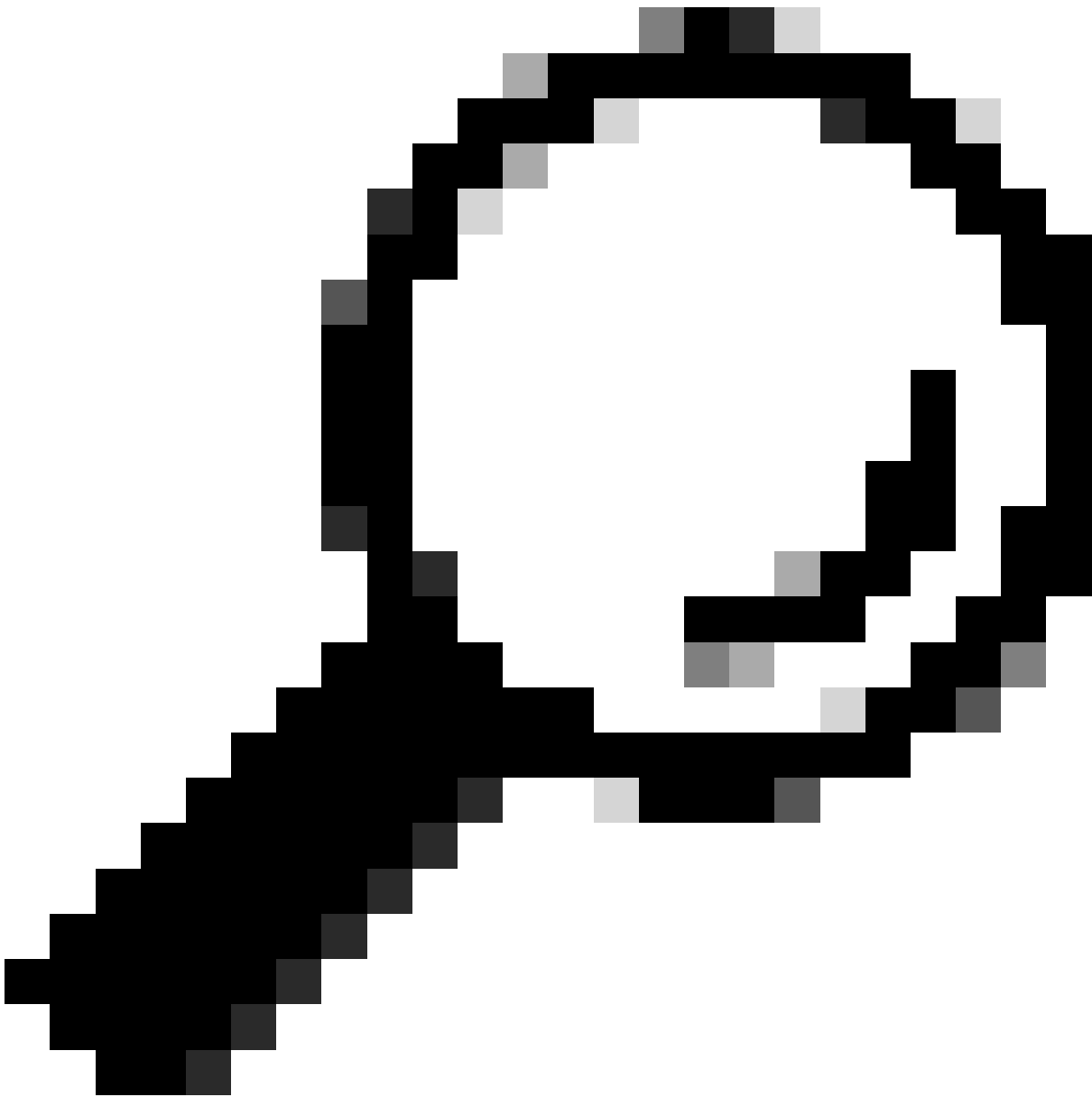
match-match een bepaalde string in de bestandsnaam, gespecificeerd als string waarde

terugkeren—Alle bestanden ophalen, inclusief submappen

met de optie comprimeren kunt u de bestanden in een gezippte indeling downloaden.



Tip: om de bestanden te downloaden, moet u ervoor zorgen dat de externe Secure File Transfer Protocol (SFTP)-server is geconfigureerd en toegankelijk is.



Tip: Met de optie Terugkeren kunt u de map voor alle subdirectory's en bestanden doorlopen. Dit wordt gebruikt als u alle logbestanden uit een map wilt halen.

Referentie links

- [JTAPI Developers Guide](#)

- [UCCX-overtrekniveaus](#)

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document ([link](#)) te raadplegen.