

Entender a arquitetura JTAPI e o fluxo de chamadas no UCCX

Contents

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Visão geral de JTAPI](#)

[JTAPI e UCCX](#)

[Arquitetura JTAPI](#)

[Modelo de observador JTAPI](#)

[Modelo de provedor JTAPI](#)

[Usuário JTAPI](#)

[Alças P1 e P2](#)

[Fluxo de chamada](#)

[Troubleshooting](#)

[Habilitar e coletar logs JTAPI](#)

[Habilitar depurações JTAPI](#)

[Coletar depurações JTAPI](#)

[Da RTMT](#)

[Do CLI](#)

[Local do log JTAPI](#)

[Links de Referência](#)

Introdução

Este documento descreve a funcionalidade básica do JTAPI, sua arquitetura e o fluxo de chamadas em relação ao UCCX.

Pré-requisitos

Requisitos

A Cisco recomenda o conhecimento destas ferramentas e recursos:

- JTAPI - API de telefonia Java
- API - Interface de programação de aplicativos

- UCCX - Unified Contact Center Express
- CUCM - Cisco Unified Communications Manager
- CTI - Integração entre telefonia e computador

A Cisco recomenda o conhecimento destes tópicos:

- Configuração do Cisco Unified Communications Manager (CUCM)
- Configuração do Unified Contact Center Express (UCCX)

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software:

- UCCX versão 12.5.1.11002-481
- CUCM versão 12.5.1.11900-146

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Informações de Apoio

Este documento descreve a arquitetura JTAPI, o fluxo de chamadas, a ativação de depurações e a coleta de logs.

Visão geral de JTAPI

O Cisco Unified JTAPI serve como um padrão de interface de programação desenvolvido pela Sun Microsystems para uso com aplicativos de telefonia por computador baseados em Java. O Cisco JTAPI implementa a especificação Sun JTAPI 1.2 com extensões adicionais da Cisco.

Você precisa usar o Cisco JTAPI para desenvolver aplicativos que:

-

Controle e observe os telefones do Cisco Unified Communications Manager.

-

Encaminhe chamadas usando portas de Integração Computador-Telefonia (CTI) e pontos de rota (dispositivos virtuais).

JTAPI e UCCX

O Cisco Unified JTAPI é usado em uma central de contatos para monitorar o status do dispositivo e emitir instruções de roteamento para enviar chamadas para o local certo na hora certa. Além disso, o JTAPI é usado para iniciar e parar a gravação de instruções enquanto recupera estatísticas de chamadas para análise e para fazer pop-ups de tela em chamadas para aplicativos CRM, scripts automatizados e controle de chamadas remotas.

Arquitetura JTAPI

O Cisco Unified JTAPI, usado em um ambiente empresarial, combina a disponibilidade, o local e as preferências do usuário para um ambiente personalizado e exclusivo para roteamento baseado em presença.

Aqui estão as aplicações da JTAPI:

- O JTAPI pode monitorar ou ser notificado sobre duas ou mais combinações de telefone e linha.
- Os aplicativos de central de atendimento usam o modelo de chamada completa para JTAPI.
- O JTAPI oferece esta funcionalidade:
 - Controle de chamadas
 - Controle de mídia
 - Negociação de mídia

Modelo de observador JTAPI

O próximo diagrama explica o modelo Provider-Observer no qual o JTAPI funciona.

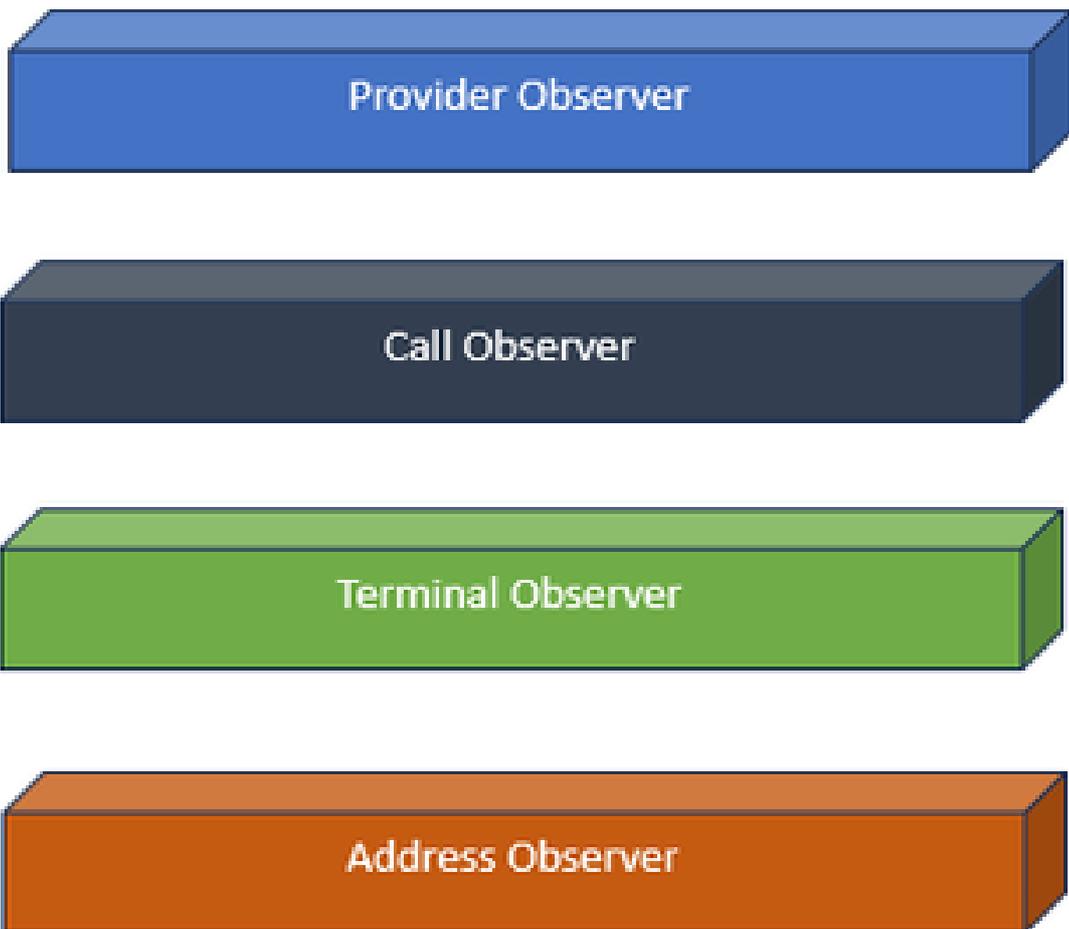
Provider

Call

Terminal

Address

Observadores



Interfaces do observador

A JTAPI baseia-se na ideia do observador. Por observador, refere-se à ideia de quem observa os eventos. Você pode ter vários observadores colocados em coisas diferentes dentro do sistema. O JTAPI usa observadores para obter informações sobre as alterações de estado dos objetos.

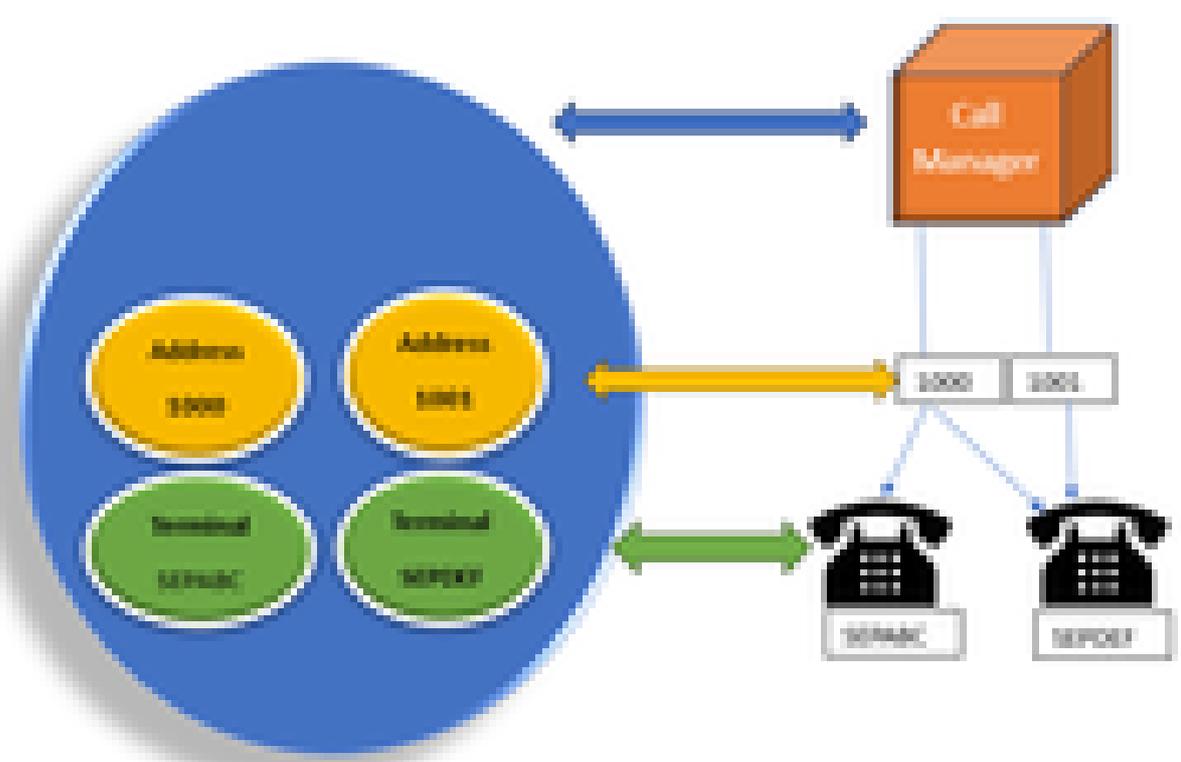
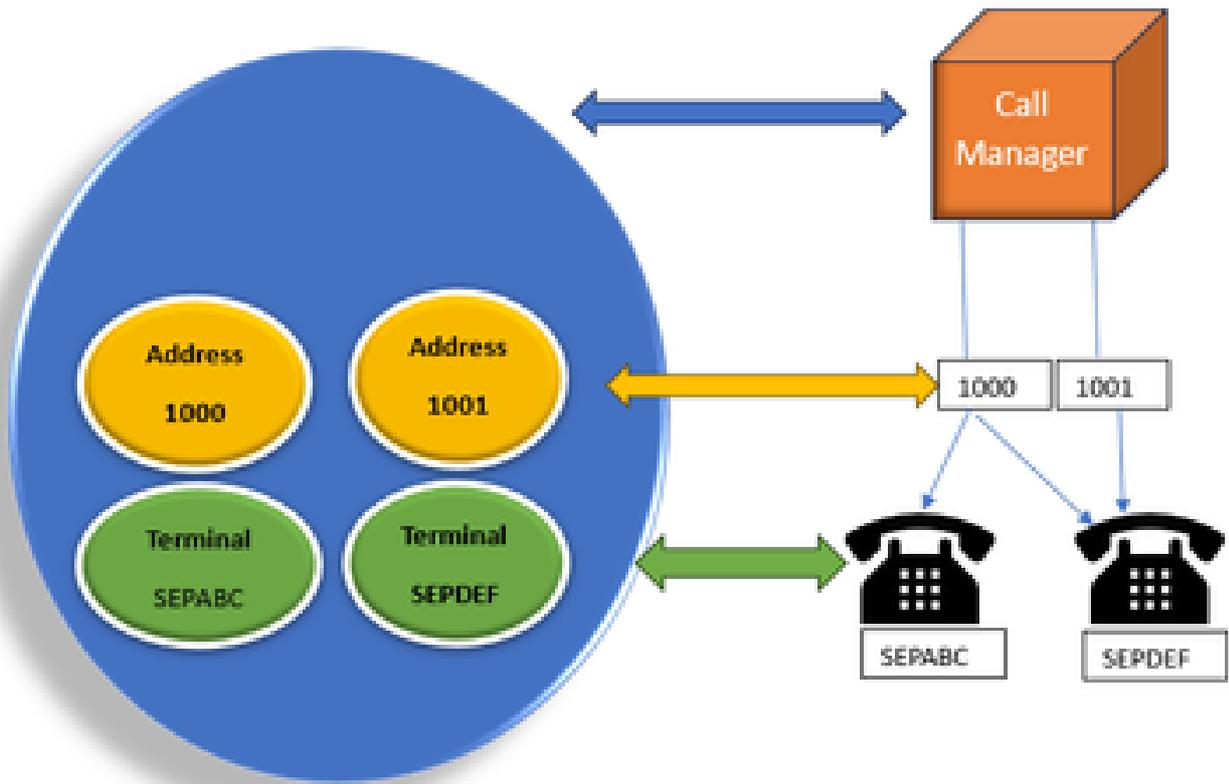
Por exemplo, você pode colocar observadores em Linhas, Telefones, terminais, endereços e assim por diante, e aprender sobre suas alterações de estado.



Observação: se não houver observadores colocados em um objeto, você não poderá ser notificado sobre as ações executadas neles.

Modelo de provedor JTAPI

O próximo diagrama explica o modelo de provedor no qual o JTAPI funciona:



Modelo
ModelProvider do

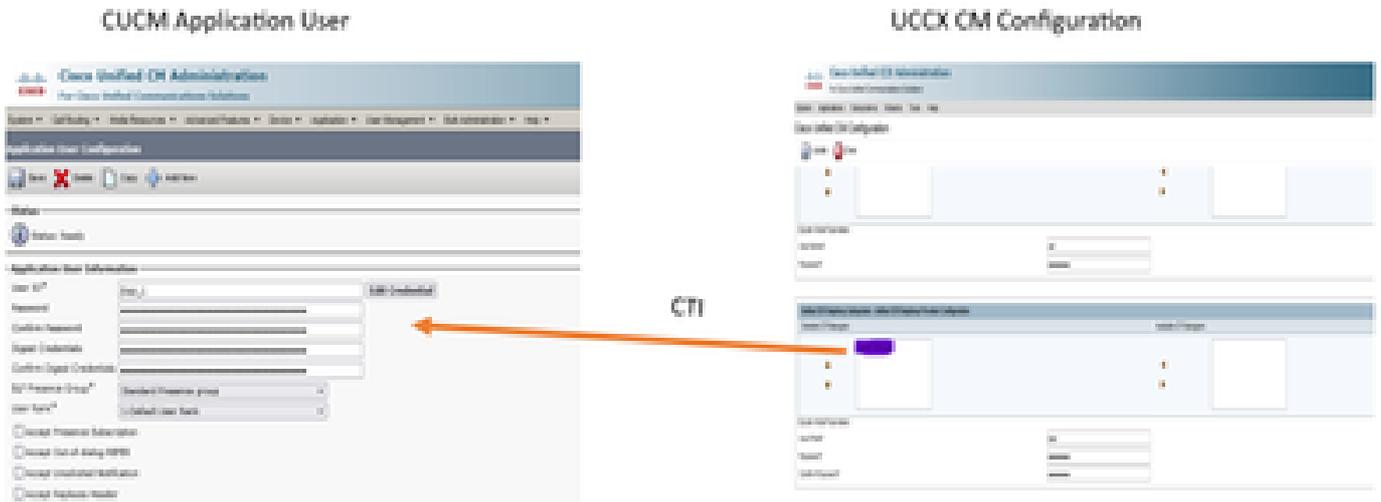
provedor JTAPI

O provedor JTAPI é uma conexão do aplicativo com o Call Manager ou o CTI Manager. Provedores são usados para anexar observadores aos

objetos. Você também pode anexar um observador a um provedor. Os provedores são usados para serem notificados sobre as ações executadas nos objetos. (Você também pode assumir o controle dos dispositivos nos quais o observador está conectado (do provedor que conectou esse observador)).

Usuário JTAPI

As próximas capturas de tela são tiradas do CUCM (esquerda) e do UCCX (direita) que explicam sobre o usuário do aplicativo JTAPI.



- Quando você inicia um aplicativo e deseja fazer uma conexão com o gerenciador CTI, você abre um provedor.
- A razão para abrir um provedor é para que você possa monitorar as ações executadas nos dispositivos, terminais e assim por diante.
- A forma como ele é implementado no CUCM é por meio de um usuário do aplicativo.
- Você cria esse usuário e fornece credenciais para que ele possa primeiro autenticar por CTI no CUCM.
- Portanto, o usuário do aplicativo JTAPI criado no CUCM é o provedor.
- Além de apenas monitorar, você precisa certificar-se de que esses dispositivos estejam em **dispositivos associados** para garantir que você possa controlar os dispositivos.



Observação: você não cria o usuário JTAPI no cucm, ele é criado pelo aplicativo (UCCX) por meio do AXL no CUCM.

Alças P1 e P2

P1 e P2 são os identificadores do provedor. Isso se torna importante quando você vai abrir vários provedores a partir do mesmo aplicativo. No UCCX, você cria 2 provedores, um dos quais controla as portas CTI e os pontos de rota, o outro controla os telefones dos agentes chamados de provedor RM-JTAPI. Você, enquanto o UCCX cria o provedor que controla as portas CTI e os pontos de rota primeiro, que é o provedor JTAPI P1. O outro provedor que você cria após o P1 é o provedor P2 ou o provedor RmCm que manipula os dispositivos do agente.

Se você vir um novo evento de chamada P1, saberá que esse é um evento de chamada de uma porta CTI ou de um ponto de rota CTI. Se você vir um evento de nova chamada P2, saberá que esse é um evento de chamada do telefone real. Você cria um usuário RM-JTAPI e um usuário JTAPI no lado CCX, mas no lado CUCM, você vê 2 usuários JTAPI criados. Isso ocorre porque cada nó do CCX recebe seu próprio usuário JTAPI,

mas ele compartilha um usuário RM-JTAPI. Quando você cria um Trigger no UCCX, ele é adicionado aos dois usuários JTAPI. Ambos os nós abrem um provedor separadamente. Assim, qualquer ação realizada no ponto de rota é notificada nos dois nós do CCX.

Fora isso, o nó secundário apenas fica lá e continua informando que ainda é o nó secundário. Cada nó tem um grupo separado de portas CTI. As portas CTI do nó 1 são controladas por jtapi_1. As portas CTI do nó 2 são controladas por jtapi_2.

Quando a chamada chega no ponto de rota CTI, ambos os nós CCX são notificados sobre o novo evento de chamada, mas o nó CCX ativo responde ao Gerenciador de chamadas com uma solicitação de redirecionamento para uma de suas portas CTI que ele controla.

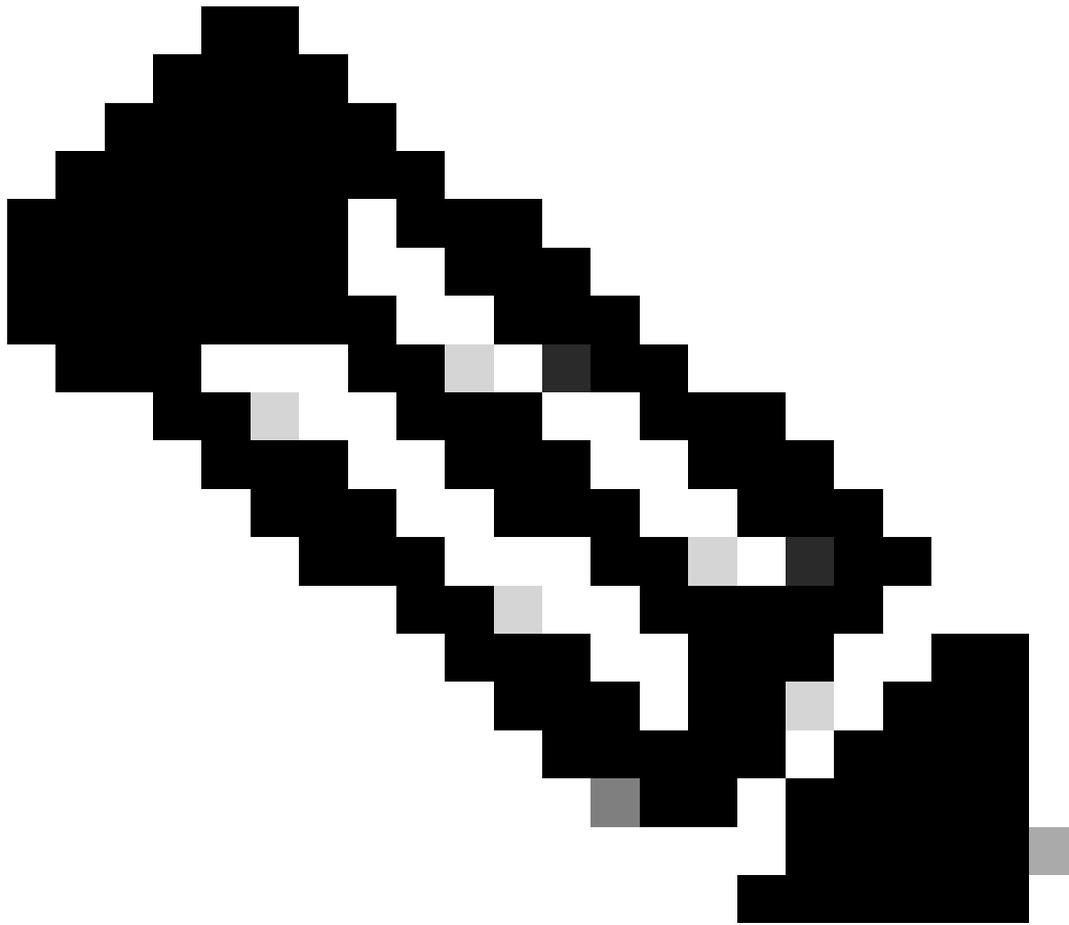
Se você vir um IP contra um ponto de rota CTI no CUCM, ele é um dos IP do CCX, mas isso não significa que o nó específico esteja roteando a chamada.

Uma coisa comum que você faz é desassociar o dispositivo do usuário JTAPI e adicioná-lo novamente. A razão por trás disso é quando você desassocia, o provedor é notificado sobre isso e remove todas as conexões a ele e, em seguida, quando é readicionado, o observador é adicionado novamente e o provedor começa a monitorá-lo novamente usando a solicitação do provedor aberto. Você pode ver que além do dispositivo que está sendo adicionado na lista de dispositivos controlados, há outra coisa chamada **Permitir controle do dispositivo do CTI** caixa de seleção no dispositivo individual também. Isso é para trazer um nível adicional de granularidade. Se o ponto de rota for adicionado à lista controlada, mas a caixa de seleção CTI não estiver marcada, você ainda poderá ser notificado sobre os eventos nesse ponto de rota, mas nenhuma ação no controle de chamadas será possível.

Fluxo de chamada

Aqui estão os seguintes de eventos envolvidos no fluxo de chamadas do UCCX:

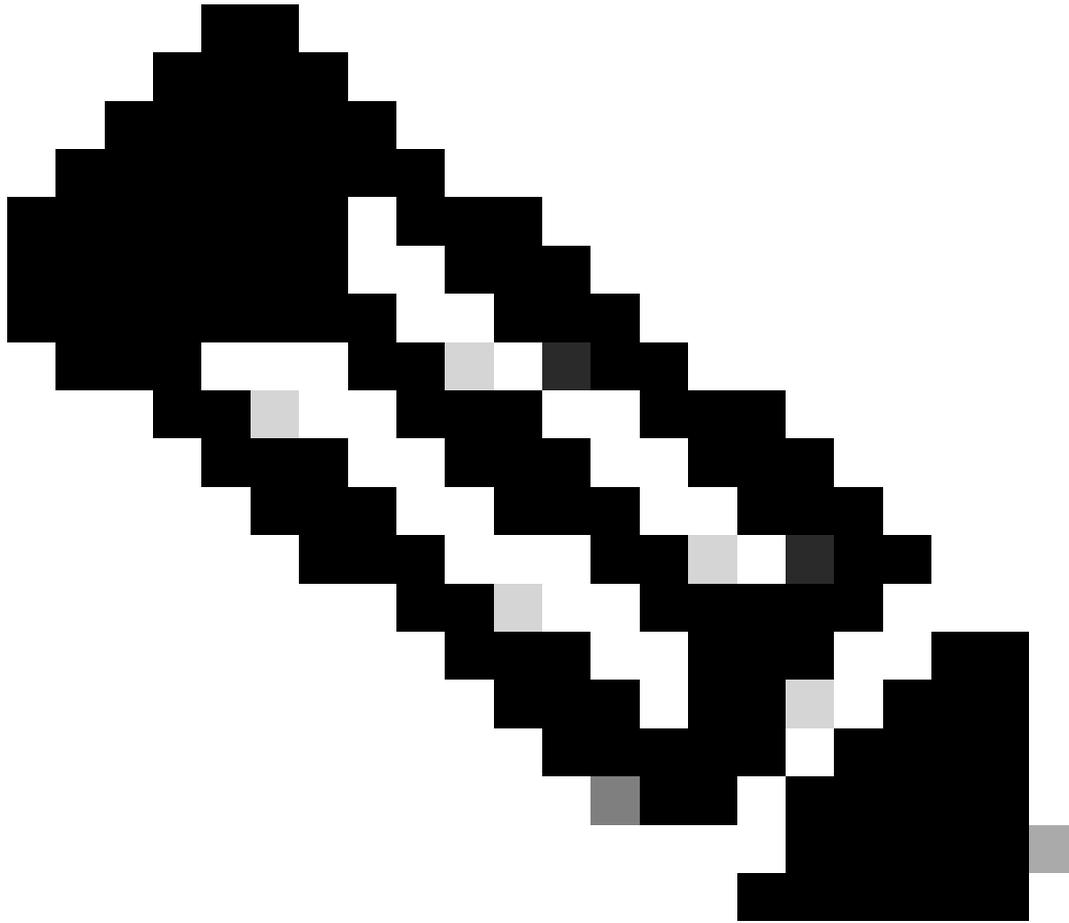
- Quando a chamada chega no ponto de rota CTI, ela é redirecionada para uma porta cti.
- A porta CTI abre o canal de mídia com o driver ipvms na caixa uccx.
- Quando você decide que o agente precisa atender a chamada, uma transferência de consulta é feita da porta para o agente.
- O novo evento de chamada é enviado ao ponto de rota CTI.
- A solicitação de redirecionamento vai para a porta CTI.
- O estado da ID da chamada se torna ocioso.
- Em seguida, outro evento de chamada novo vem para a chamada para a porta CTI.
- Se a resposta de redirecionamento for 0, ela estará boa; se for diferente de zero, significa que ela falhou.
- Em seguida, você envia a solicitação de aceitação de chamada (ela não é respondida, apenas aceita na porta).
- Em seguida, aceite as ocorrências de etapa no script, isto é, quando a solicitação de atendimento de chamada vem em Jtapi.



Observação: isso acontece tão rápido e, às vezes, vários eventos ocorrem ao mesmo tempo, como chamadas provenientes do Cisco Unity Connection ou transferências que levam tempo do CUCM, isso pode causar uma condição RACE na etapa de aceitação, que é o motivo pelo qual você deseja reduzir essa velocidade. Para fazer isso, adicione a etapa de atraso antes da etapa de aceitação.

11. Em seguida, você recebe uma resposta da porta.

12. O estado da chamada é alterado para conectado.



Observação: o CTI é assíncrono, ao contrário dos telefones sip ou skinny, que aguardam a resposta antes de enviar uma nova solicitação. É por isso que a ordem das mensagens nas mensagens JTAPI CTI pode ser embaralhada.

13. Depois de obter a resposta da porta, ocorre a negociação da mídia.

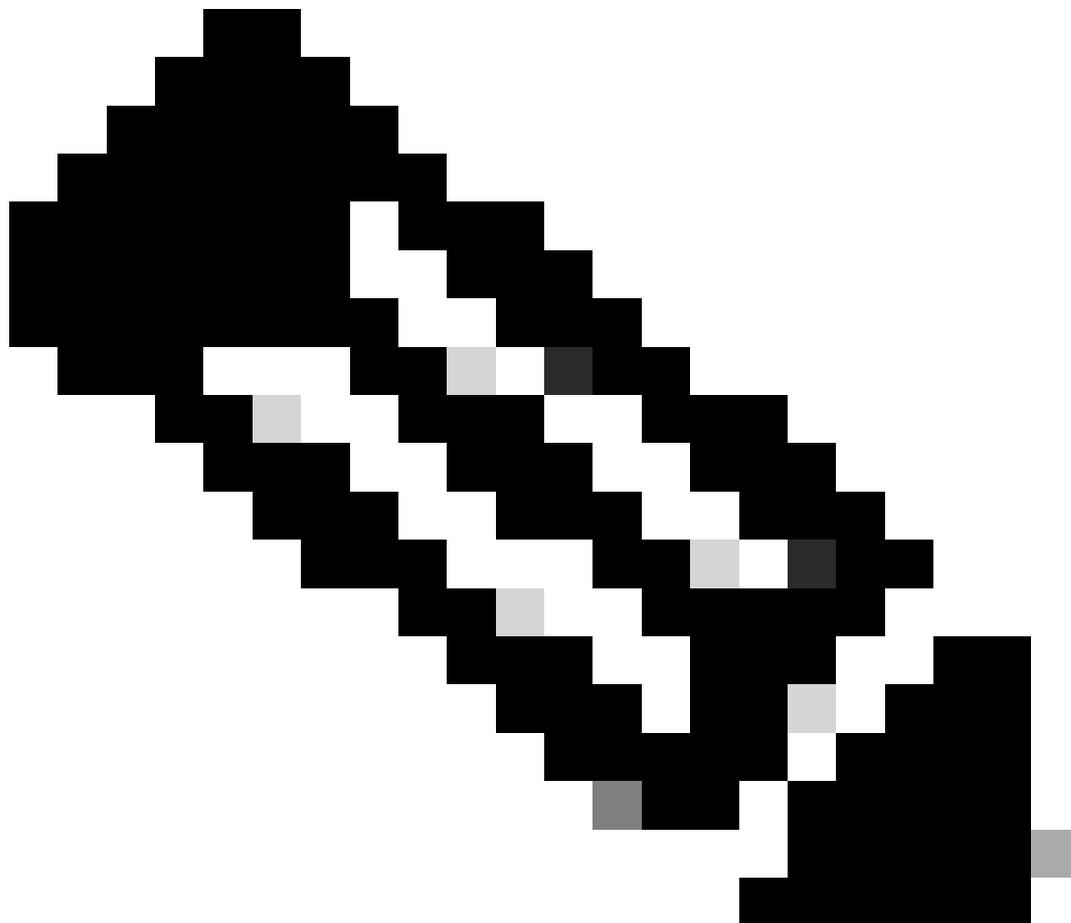
14. A porta envia a solicitação `open_logical_channel` na qual ela compartilha sua porta e ip para os quais ela deseja que a parte remota envie o RTP.

15. Antes de abrir o canal lógico, ele cria uma conexão com o driver IPVMS na caixa UCCX e abre um fluxo RTP.

16. O próximo evento é o evento `start_receive`, que nos informa as informações da extremidade oposta (ou seja, o ip e a porta do dispositivo chamador).

17. Em seguida, você obtém o evento `start_transmission` como qualquer outro sinal de mídia.

18. Agora você está ouvindo a IVR e os prompts.



Observação: é aqui que a configuração da chamada é concluída.

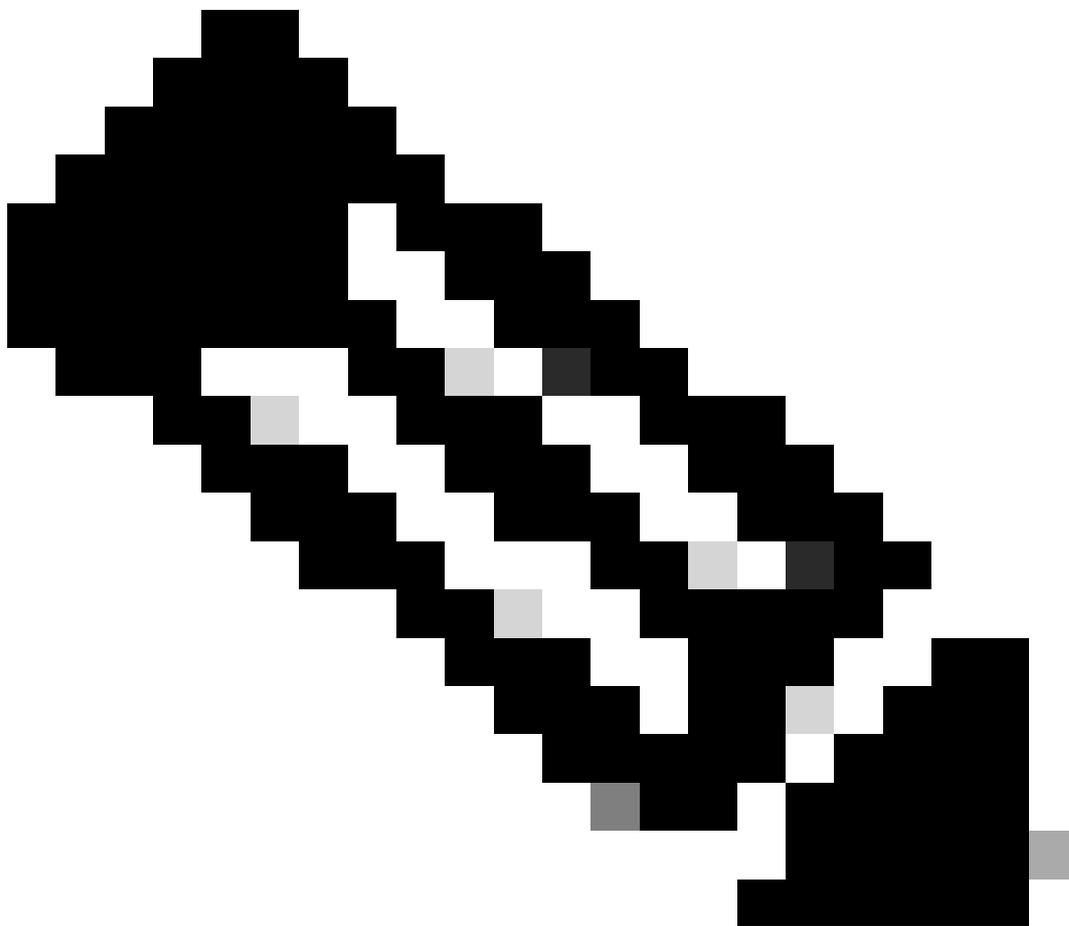
19. Agora, quando a chamada atinge uma etapa no script que permite a transferência da chamada para o agente, você vê uma `CallSetupTransferRequest`.

20. Ao contrário da primeira mensagem, esta é uma transferência de consulta e não um redirecionamento.

21. A razão para isso ser uma transferência de consulta é que se um agente estiver PRONTO, mas não em seu assento, e redirecionarmos a chamada, ela simplesmente permanecerá sem resposta, mas se fizermos uma transferência de consulta, se o agente não estiver lá, a chamada será

colocada novamente na fila.

22. Assim como qualquer outra transferência de consulta, essa é a porta CTI que pressiona o botão de transferência pela primeira vez no telefone.



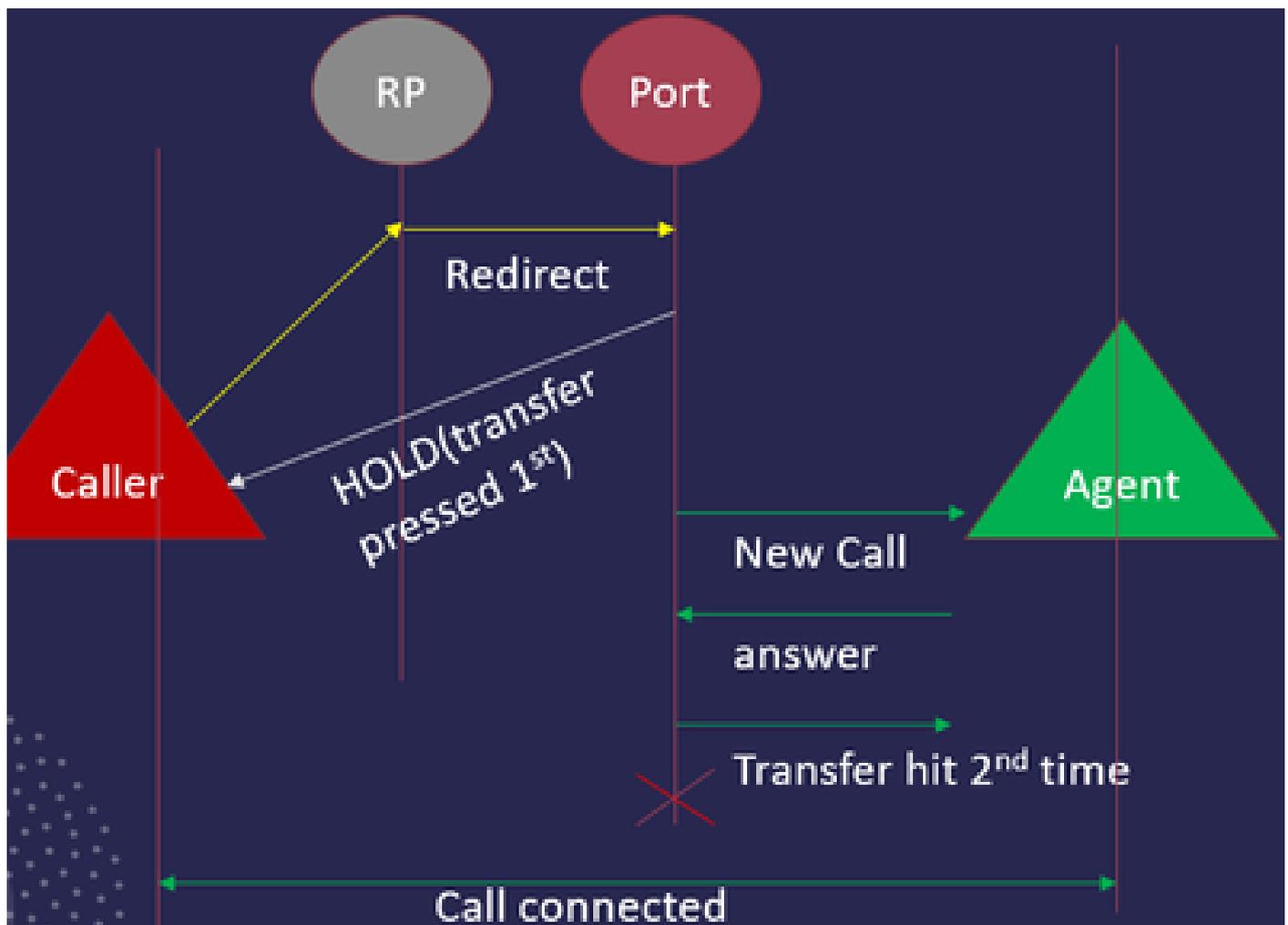
Observação: ConsultWithoutMedia é a API para a transferência de consulta.

23. Em uma transferência de consulta regular, há um caminho de mídia entre A e C, mas nesse caso você instrui o CUCM a não fazer isso, pois não faz sentido que você estabeleça a mídia entre o UCCX e o Agente.

24. Então você vai fazer uma transferência de consulta sem fazer um corte de mídia entre o agente e a porta CTI.

25. Nesse ponto, a porta CTI coloca o chamador em espera e vemos um evento de estado de chamada alterado=ESPERA.
26. Agora você verá um evento de nova chamada da porta CTI para o dispositivo do agente.(Usando a ID de chamada original, mas um subconjunto dela e um evento P2.)
27. Se você pesquisar a ID de chamada do evento P2, verá a próxima mensagem como CallAnswer request (solicitação de resposta à chamada), o que significa que o agente atendeu a chamada.
28. Assim que você souber que o agente atendeu a chamada (usando P2) e que a chamada também está no estado conectado na porta CTI (usando P1), então você verá um CallDirectTransferRequest Ponto de Rota, o que significa que o botão de transferência foi pressionado pela segunda vez.
29. Agora que a porta CTI sabe que o agente atendeu a chamada, não há nenhum ponto de espera, ela imediatamente envia uma CallDirectTransferRequest que é a porta CTI que pressiona o botão de transferência segunda vez, conforme explicado acima.
30. Agora, a perna de chamada original (a que estava em espera) é a que sobreviveu.
31. O outro trecho da chamada é destruído (entre a porta e o agente).
32. Neste ponto, o CUCM e o UCCX saem da imagem e o RTP é estabelecido entre o chamador e o agente através do Gateway.

O próximo diagrama explica o fluxo de chamadas mencionado anteriormente de forma resumida.



Troubleshooting

Habilitar e coletar logs JTAPI

Habilitar depurações JTAPI

Verifique estas etapas para habilitar os níveis de depuração JTAPI.

- Faça login no UCCX.
- Vá para **CCX Serviceability**.
- Vá para **Trace**.
- Vá para **Configuração**.
- No menu suspenso Selecionar serviço, selecione **Cisco Unified CM Telephony Client**.
- Marque a caixa de seleção **Depuração**.
- Selecione todas as caixas de seleção, exceto **MISC_DEBUGGING**.

Coletar depurações JTAPI

Verifique estas etapas para habilitar os níveis de depuração JTAPI de RTMT ou CLI.

Da RTMT

- Faça login na Ferramenta de monitoramento em tempo real do CCX.
- Clique em **Trace & Log Central**.
- Clique em **Coletar arquivos**.
- Selecione **JTAPI Client** para todos os servidores.
- Clique em Next.
- Selecione os carimbos de data/hora e o local em que deseja salvar os arquivos de log.

Do CLI

Local do log JTAPI

ativelog /uccx/log/JTAPI

Comando para coletar os logs JTAPI:

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

Sintaxe:

```
file get { ativelog|inativelog|install } file-spec [opções]
```

arquivo de especificação de arquivo obrigatório a ser transferido

opções reltime opcional meses|semanas|dias|horas|minutos valor de tempo

abstime hh:mm:MM/DD/AA hh:mm:MM/DD/AA

match regex

recorrências

compactar

5 maneiras de fazer download dos logs com base no timestamp

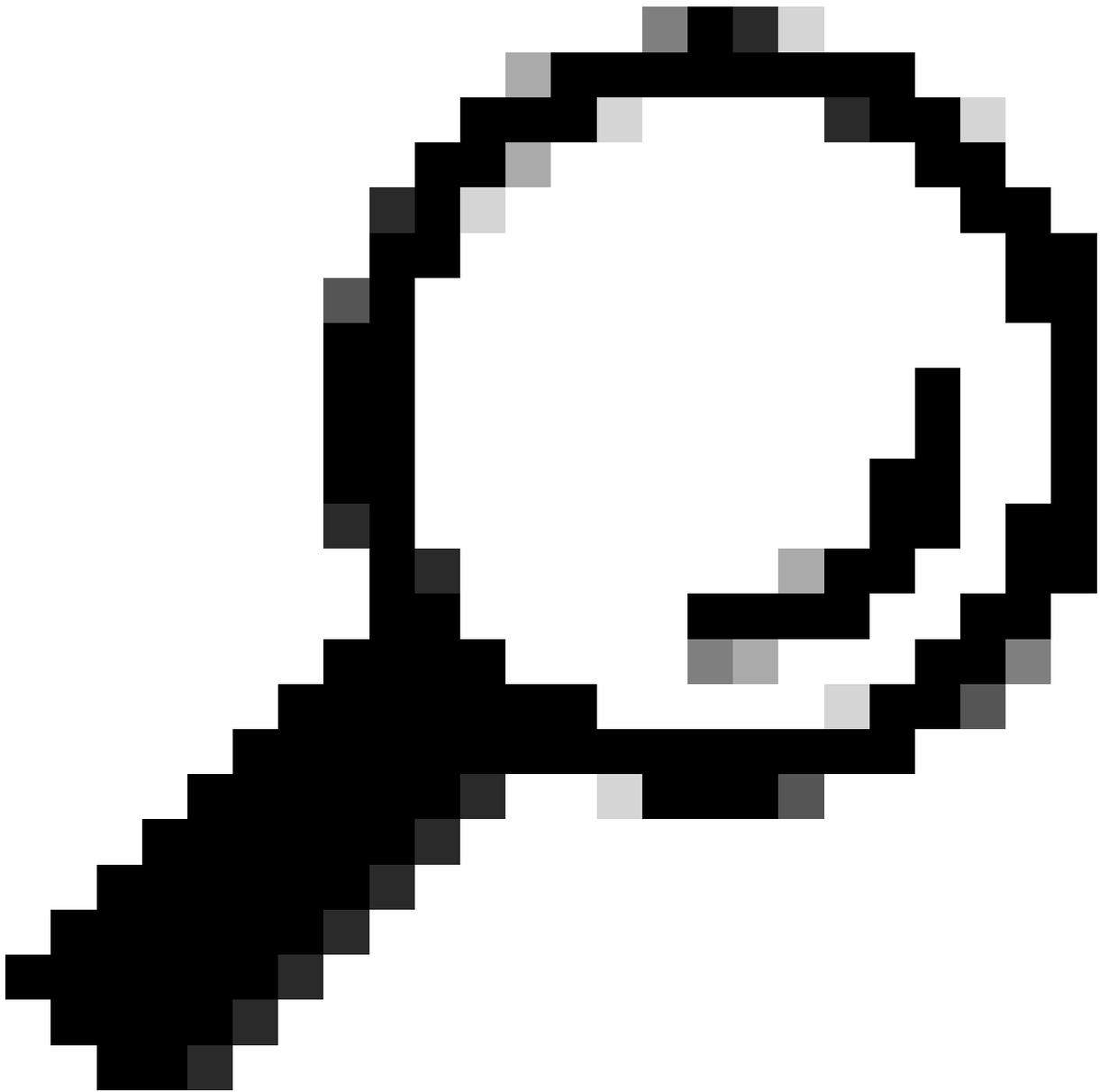
reltime — Período de tempo relativo, especificado como minutos | horas | dias | semanas | valor de meses

abstime — Período de tempo absoluto, especificado como hh:mm:MM/DD/YY hh:mm:MM/DD/YY

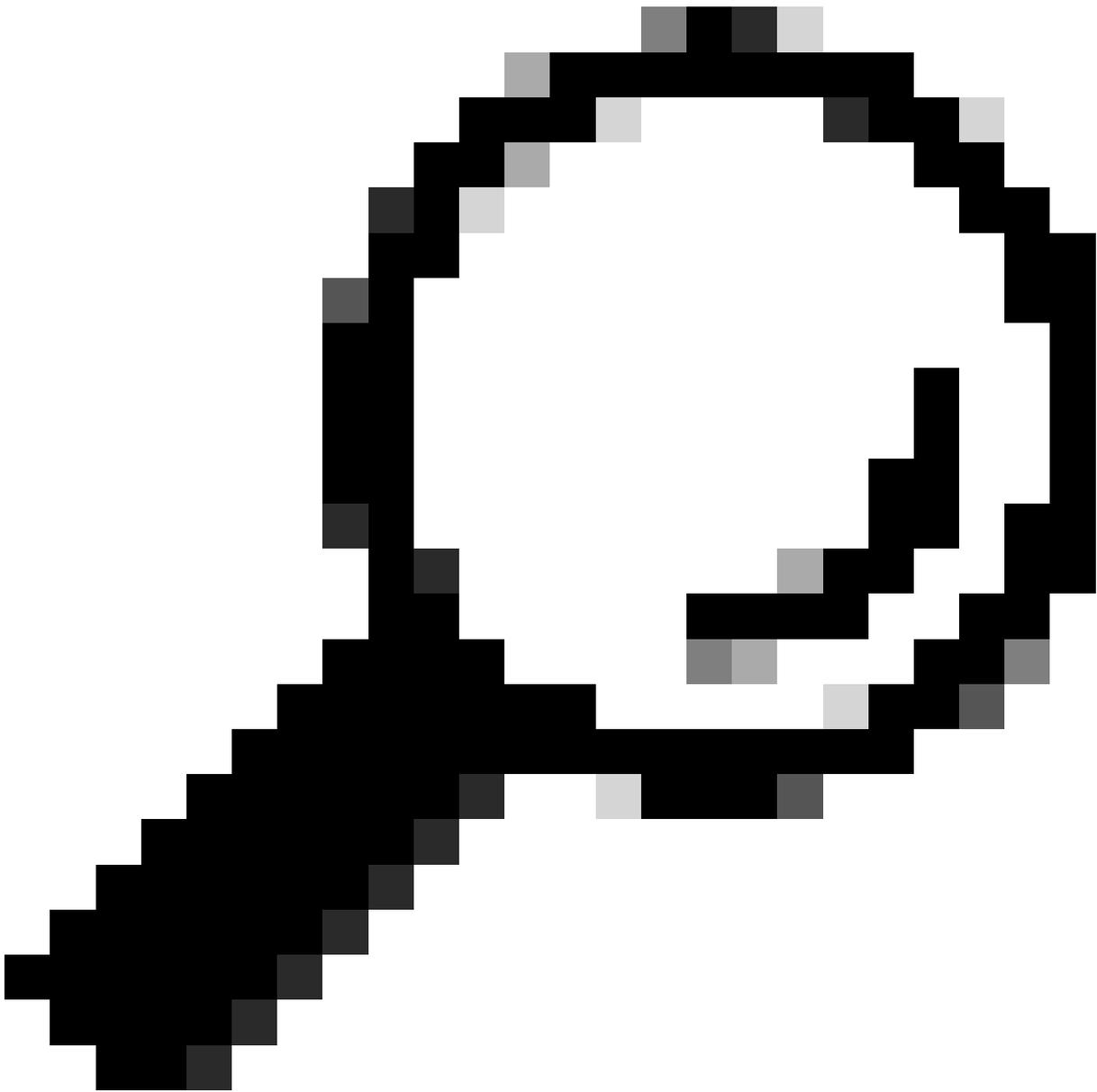
match — Corresponde a uma string específica no nome do arquivo, especificado como valor da string

recurs — Obter todos os arquivos, inclui subdiretórios

compress option permite que você faça o download dos arquivos em um formato zipado.



Dica: para baixar os arquivos, certifique-se de que o servidor SFTP externo esteja configurado e acessível.



Dica: a opção `recurs` permite que você percorra o diretório para todos os subdiretórios e arquivos. Isso é usado se você quiser receber todos os logs de um diretório.

Links de Referência

- [Guia para desenvolvedores de JTAPI](#)

- [Níveis de rastreamento do UCCX](#)

Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.