ダイヤル技術接続のトラブルシューティング - IOS DDR初期コール

目次

概要

このドキュメントでは、さまざまなダイヤル接続のタイプのトラブルシューティング方法を提供しており、最初から最後まで読むことを目的とはしていません。 この構成は、読者が関心のある項に進めるように設計されており、それぞれが特定のケースの全体的なトラブルシューティングのテーマのバリエーションになっています。

前提条件

要件

このドキュメントでは、3 つの主要なシナリオについて説明します。 トラブルシューティングを開始する前にどのようなタイプのコールが試みられているか特定し、対応する項に移動してください。

- callin
- Cisco IOS ダイヤルオンデマンド ルーティング (DDR)
- 非 DDR 呼び出し

使用するコンポーネント

このドキュメントは、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。 このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。 対象のネットワークが実稼働中である場合には、どのような作業についても、その潜在的な影響について確実に理解しておく必要があります。

履歴

ダイヤルアップは、端末ユーザのためにデータを搬送する Public Switched Telephone Network（PSTN; 公衆電話交換網）のための機能です。 ダイヤルアップには、接続の宛先となる電話番号を電話交換機に送信する Customer Premises Equipment（CPE; 顧客宅内機器）デバイスが含まれます。 AS3600、AS5200、AS5300、AS5800 などは、一次群速度インターフェイス（PRI）を一連のデジタル モデムとともに実行できるルータです。 一方、AS2511 は外部モデムと通信するルータです。

現在、通信事業の市場ではその規模の拡大とともに、より高いモデム密度が求められています。 このニーズへの答えは、電話会社の機器とのインターオペラビリティの向上と、デジタル モデムの開発です。 デジタル モデムには PSTN に直接デジタル アクセスする機能があります。 結果として、現在ではデジタル モデムの特長である信号の明瞭さを利用した、より高速な CPE モデムが開発されています。 デジタル モデムは PRI または基本速度インターフェイス（BRI）を通じて PSTN に接続し、V.90 通信規格を使用して 53k 超のデータを伝送できるので、理想を実現します。

初めて市場に投入されたアクセス サーバは AS2509 および AS2511 でした。 AS2509 は外部モデムを使用して 8 つの着信接続を処理でき、AS2511 は 16 の着信接続を処理できました。 2 つの PRI が導入された AS5200 はデジタル モデムを使用して 48 のユーザをサポート可能であり、収容効率が飛躍的に向上しました。 AS5300 では PRI のサポートが 4 から 8 に増え、モデム密度が着実に増加していきました。 最終的に、数十の T1 回線と数百のユーザ接続を処理する通信事業者クラスの設置ニーズに対応するため、AS5800 が導入されました。

ダイヤラ テクノロジーの歩みの中で、現在でも議論の対象にされる旧来のテクノロジーがいくつかあります。 56Kflex は、Rockwell によって提唱された（V.90 以前の）古い 56k モデム規格です。 Cisco では、バージョン 1.1 の 56Kflex 規格を Cisco 製の内部モデムでサポートしていますが、CPE をできるだけ速やかに V.90 に移行することを推奨しています。 もう 1 つの旧来のテクノロジーとして、AS5100 があります。 AS5100 は Cisco とモデム製造元が共同で開発した製品でした。 AS5100 は、クワッド モデム カードを使用してモデム密度を増やす手段として開発されました。 AS5100 ではカードとして組み込まれた AS2511 が必要でした。これらのカードは、クワッド モデム カードとデュアル T1 カードによって共有されるバックプレーンに挿入されていました。

表記法

ドキュメント表記の詳細は、『シスコ テクニカル ティップスの表記法』を参照してください。

Cisco IOS DDR によるコール開始

着信コールのトラブルシューティングが下位層からアプローチを始めるのに対して、発信接続のトラブルシューティングは上位層からアプローチを始めます。

論理の全般的なフローは次のようになります。

1. ダイヤルオンデマンド ルーティング（DDR）がコールを開始しましたか。（答えがはいの場合は、次の質問に進みます）。
2. 非同期モデムの場合、チャット スクリプトから期待されるコマンドが発行されましたか。
3. コールが PSTN の外部に達しましたか。
4. リモート エンドがコールに応答しましたか。
5. コールが完了しましたか。
6. データがリンクを通過していますか。
7. セッションが確立されましたか。（PPP または端末）

ダイヤラがリモートの宛先にコールを発信しようとしているかどうかを調べるには、**debug dialer events** コマンドを使用します。 詳細については **debug dialer packet** を参照してください。ただし、**debug dialer packet コマンド**はリソースの消費が激しいため、ダイヤラ インターフェイスが複数動作しているビジー状態のシステムでは使用しないでください。

次の行は IP パケットに対する **debug dialer events** の出力行で、DDR インターフェイスの名前と、パケットの送信元アドレスおよび宛先アドレスが列挙されています。

```
Dialing cause: Async1: ip (s=172.16.1.111 d=172.16.2.22)
```

トラフィックによってダイヤルが開始されない場合、最も可能性のある理由として不適切な設定（対象トラフィックの定義、ダイヤラ インターフェイスの状態、またはルーティングのいずれか）が考えられます。

表 1：トラフィックによってダイヤル試行が開始されない場合

考えられる原因	推奨される対策
「対象トラフィック」の定義がない、または正しくない	1. **show running-config** コマンドを使用し、インターフェイスに **dialer-group** が設定されていること、および一致する番号で設定されたグローバル レベルの **dialer-list** が存在することを確認します。 2. **dialer-list** コマンドが、プロトコル全体、またはアクセス リストに一致するトラフィックのどちらかを許可するように設定されていることを確認します。 3. **アクセス リスト**で、リンクを通過するパケットが「対象」として宣言されていることを確認します。 これをテストするには、該当するアクセス リストの番号を指定して特権 exec コマンド **debug ip packet [list number]** を使用すると便利です。 次に、リンクに対して ping を試すか、またはトラフィックを送信します。 対象トラフィック フィルタが正しく定義されていれば、パケットがデバ

	ッグ出力に表示されます。 このテストでデバッグ出力が表示されない場合は、**アクセスリスト**とパケットが一致していません。
インターフェイスの状態	**show interfaces <interface name>** コマンドを使用し、インターフェイスが「アップ/アップ（スプーフィング）」状態にあることを確認します。
• インターフェイスが「スタンバイ」モードにある	ルータのもう 1 つの（プライマリ）インターフェイスが、ダイヤラ インターフェイスをバックアップ インターフェイスとして使用するように設定されています。 また、プライマリ インターフェイスが「ダウン/ダウン」の状態ではありません。ダイヤラ インターフェイスがスタンバイ モードから抜け出すためには「ダウン/ダウン」状態にあることが必要です。 さらに、プライマリ インターフェイスで **backup delay** を設定する必要があります。これが設定されていないと **backup interface** コマンドが実行されません。 ダイヤラ インターフェイスが「スタンバイ」から「アップ/アップ（スプーフィング）」に変わることを確かめるには、通常、プライマリ インターフェイスからケーブルを抜き取る必要があります。 設定コマンド shutdown を使用してプライマリ インターフェイスをシャットダウンしただけでは、プライマリ インターフェイスは「ダウン/ダウン」にはならず、「管理上のダウン」になります。 これらは同じではありません。 加えて、プライマリ接続がフレームリレー経由の場合、フレームリレーの設定をポイントツーポイントのシリアル サブインターフェイスで行う必要があり、電話会社で「Active」ビットを渡している必要があります。 この方法を「エンドツーエンド ローカル管理インターフェイス（LMI）」とも呼びます。
• インターフェイスが「administratively down」にある	ダイヤラ インターフェイスに **shutdown** コマンドが設定されています。 Cisco ルータを初めてブートしたときは、どのルータ インターフェイスでもこの状態がデフォルトです。 これを解除するには、インターフェイス設定コマンド **no shutdown** を使用します。
ルーティングに誤りがある	exec コマンド show ip route [a.b.c.d] **を発行します。ここで** a.b.c.d **は、リ**

	モート ルータのダイヤラ インターフェイスのアドレスです。 リモート ルータで ip unnumbered を**使用している**場合は、ip unnumbered コマンドでリストされるインターフェイスのアドレスを使用します。 出力には、ダイヤラ インターフェイスを経由したリモート アドレスへのルートが示されます。 ルートが存在しない場合は、show running-config の出力を調べて、スタティック ルートまたはフローティング スタティック ルートがすでに設定されていることを確認します。 ダイヤラ インターフェイス以外のインターフェイスを経由するルートが**存在する場合**は、DDR がバックアップとして使用されています。 ルータの設定を調べ、スタティック ルートまたはフローティング スタティック ルートがすでに設定されているかを確認します。 この場合にルーティングをテストするには、プライマリ接続を無効にし、**show ip route [a.b.c.d] コマンドを実行して、適切なルートがルーティング テーブルに設定されていることを確認する**のが最も確実な方法です。 **注:** この操作をネットワークの稼働中に行うと、ダイヤル イベントがトリガーされることがあります。 この種のテストはあらかじめスケジューリングされたメンテナンス時に行うのが最適です。

コールの送信

ルーティングと対象トラフィック フィルタが正しければ、コールが開始されます。 これは **debug dialer events** を使用して確認できます。

```
Async1 DDR: Dialing cause ip (s=10.0.0.1, d=10.0.0.2)
```

```
Async1 DDR: Attempting to dial 5551212
```

ダイヤリング理由が表示されるにもかかわらず、ダイヤル試行がなされない場合、通常はダイヤラ マップまたはダイヤラ プロファイルの設定の誤りが原因です。

表 2： コールが発信しない場合

考えられる問題	推奨する処置
ダイヤラ マップの設定の誤り	**show running-config** コマンドを使用し、ダイヤリング インターフェイスに少なくとも 1 つの **dialer map** 文が設定されてい

	て、この文がリモート サイトのプロトコル アドレスと送信先番号を指し示していることを確認します。
ダイヤラ プロファイルの設定の誤り	**show running-config** コマンドを使用し、Dialer インターフェイスに **dialer pool X コマンド**が設定されていて、対応する **dialer pool-member X** がルータのダイヤラ インターフェイスに設定されていることを確認します。 ダイヤラ プロファイルが正常に設定されていない場合、次のようなデバッグ メッセージが表示されることがあります。 `Dialer1: Can't place call, no dialer pool set`**dialer string** が設定されていることを確認します。

次に、使用されているメディアのタイプを確認します。

- 外部非同期モデム DDR
- 個別線信号方式（CAS）T1/E1 非同期モデム DDR
- PRI 非同期モデム DDR
- BRI 非同期モデム DDR
- PRI ISDN DDR
- BRI ISDN DDR

外部非同期モデム DDR

1. 外部非同期モデム DDR を確認するには、次のコマンドを使用してから、コール発信を試行します。`router# debug modem router# debug chat line <n>`
2. モデム コールの場合、コールを続行するためチャット スクリプトが実行される必要があります。 ダイヤラ マップ ベース DDR の場合、チャット スクリプトはダイヤラ マップ コマンドの **modem-script パラメータ**によって起動されます。 DDR がダイヤラ プロファイル ベースの場合は、TTY 回線で設定されたコマンド script **dialer** によって行われます。 どちらを使用する場合も、ルータのグローバル コンフィギュレーション内に存在する次のようなチャット スクリプトを利用します。`chat-script callout AT OK atdt\T TIMEOUT 60 CONNECT \c` いずれの場合でも、チャット スクリプトのアクティビティを表示するコマンドは **debug chat** です。 たとえば **dialer map または dialer string コマンド**内のダイヤル文字列（電話番号）が 5551212 であれば、デバッグ出力は次のようになります。`CHAT1: Attempting async line dialer script`

```
CHAT1: Dialing using Modem script: callout & System script: none
CHAT1: process started
CHAT1: Asserting DTR
CHAT1: Chat script callout started
CHAT1: Sending string: AT
CHAT1: Expecting string: OK
CHAT1: Completed match for expect: OK
CHAT1: Sending string: atdt5551212
CHAT1: Expecting string: CONNECT
CHAT1: Completed match for expect: CONNECT
CHAT1: Chat script callout finished, status = Success
```

3. チャット スクリプトが、「Sending string」に基づいて予期されるコール（正しい番号）を

試行することを確認します。 チャット スクリプトが予期されるコールを発信しない場合は、チャット スクリプトの設定を確認します。 exec プロンプトで **start-chat コマンド**を使用して、チャット スクリプトを手動で開始します。

4. 「Timeout expecting: CONNECT」には、さまざまな原因が記述されています。**原因 1：** ローカル モデムが実際にコールを発信していません。 モデムへのリバース Telnet を実行してダイヤルを手動で開始し、モデムがコールを発信できることを確認します。 リモート エンドが応答しない場合は、発信元モデルからコールが発信されていることを確認するため、**ATDT <number>** コマンドを使用してローカル番号に手動で発信し、呼出音を聞きます。**原因 2：** リモート モデムが応答していません。 通常の POTS 電話を使用してリモート モデムにダイヤリングし、動作をテストします。 次の手順を試してください。コールした電話番号が正しいことを確認します。 ハンドセットを使用して着信番号をコールします。 モデムに使用していたものと同じ壁面接続ケーブルを使用していることを確認します。 手動コールが着信非同期モデムに到達できる場合は、発信元モデムが正しく機能していない可能性があります。 モデムを確認し、必要に応じて交換します。手動コールが応答非同期モデムに到達できない場合は、着信側モデムの電話ケーブルを取り換えてから、着信側モデムの回線で通常の電話機を使って試してみます。 通常の電話機でコールが受信できる場合は、着信側モデムに問題がある可能性があります。手動コールが当該回線で通常の電話機に到達しない場合は、着信側施設の別の（確認済みの良好な）回線を使用してみます。 正常に接続したら、着信側モデムに接続する電話回線の検査を電話会社に依頼します。長距離コールの場合、発信元で別の（確認済みの良好な）長距離番号を試してみます。 この番号で通話できる場合は、着信側施設または回線が長距離コールを受信するようにプロビジョニングされていない可能性があります。 発信側回線が他の長距離番号に到達できない場合は、長距離コールが有効になっていない可能性があります。 別の長距離通話会社で 1010 コードを試行します。最後に、発信元回線から（確認済みの良好な）別のローカル番号を試行します。 それでも接続が失敗する場合は、発信元回線の検査を電話会社に依頼します。**原因 3：** ダイヤルしようとしている番号に誤りがあります。 手動でダイヤリングを行い、番号を確認します。 必要であれば設定を訂正します。**原因 4：** モデムの trainup に時間がかかりすぎているか、または TIMEOUT の値が小さすぎます。 **chat-script コマンド**で TIMEOUT 値を大きくしてみます。 TIMEOUT がすでに 60 秒以上の場合は、モデムと、モデムに接続している DTE の間のケーブルに問題がある可能性があります。 trainup が失敗する場合は、回線の問題、またはモデムに互換性がないことが考えられます。個々のモデムの問題の根本的な原因を突き止めるには、reverse telnet を使用して発信元モデムで AT プロンプトに移動します。可能であれば、着信側モデムでも AT プロンプトに移動します。 ほとんどのモデムは、各自の DTE 接続に接続された端末セッションの呼び出しを示します。 ATM1 を使用して、モデムがそのスピーカに音声を送信するようにします。これにより、回線で何が起こっているかを両側で聞くことができます。音楽に雑音が混じっているかを確認します。 雑音が混じっている場合は、回線をクリーンアップします。 非同期モデムが trainup していない場合は、番号に発信してスタティックを聞きます。 他の要素が trainup と干渉している可能性があります。 非同期モデムへリバース Telnet を実行してデバッグします。
5. すべてが正しく動作しているものの、CAS 非同期モデム DDR で接続できない場合は、PPP デバッグを試行します。 次のコマンドを使用します。`router# debug ppp negotiation router# debug ppp authentication` チャット スクリプトが正常に完了すると、モデムが接続されます。 17 PPP

CAS T1/E1 非同期モデム DDR

1. CAS T1/E1 非同期モデム DDR を確認するには、次のコマンドを使用してから、コール発信

を試行します。**警告：** *ビジー状態のシステムでデバッグを実行すると、CPU への過負荷またはコンソール バッファのオーバーランが原因でルータがクラッシュすることがあります*。router# **debug modem** router# **debug chat** or **debug chat line** *n* router# **debug modem csm** router# **debug cas** **注: debug cas コマンドは、**Cisco IOS?? ソフトウェア リリース 12.0(7)T 以降が稼働する Cisco AS5200 および AS5300 プラットフォームで使用できます。 以前のバージョンの IOS では、**service internal** コマンドをルータのコンフィギュレーションのメイン レベルに入力し、**modem-mgmt csm debug-rbs** を exec プロンプトで入力する必要がありました。 Cisco AS5800 でデバッグを行うには、トランク カードに接続する必要があります。（デバッグをオフにするには **modem-mgmt csm no-debug-rbs** を使用します。）

2. モデム コールの場合、コールを続行するためチャット スクリプトが実行される必要があります。 ダイヤラ マップ ベース DDR の場合、チャット スクリプトはダイヤラ マップ コマンドの **modem-script パラメータ**によって起動されます。 DDR がダイヤラ プロファイル ベースの場合は、TTY 回線で設定されたコマンド script **dialer** によって行われます。 どちらを使用する場合も、ルータのグローバル コンフィギュレーション内に存在する次のようなチャット スクリプトを利用します。chat-script callout AT OK atdt\T TIMEOUT 60 CONNECT \c いずれの場合でも、チャット スクリプトのアクティビティを表示するコマンドは **debug chat** です。 たとえば **dialer map または dialer string コマンド**内のダイヤル文字列（電話番号）が 5551212 であれば、デバッグ出力は次のようになります。CHAT1: Attempting async line dialer script

```
CHAT1: Dialing using Modem script: callout & System script: none
CHAT1: process started
CHAT1: Asserting DTR
CHAT1: Chat script callout started
CHAT1: Sending string: AT
CHAT1: Expecting string: OK
CHAT1: Completed match for expect: OK
CHAT1: Sending string: atdt5551212
CHAT1: Expecting string: CONNECT
CHAT1: Completed match for expect: CONNECT
CHAT1: Chat script callout finished, status = Success
```

3. チャット スクリプトが「Sending string」に基づいて予期されるコール（正しい番号）を試行することを確認します。 チャット スクリプトが予期されるコールを発信しない場合は、チャット スクリプトの設定を確認します。 exec プロンプトで **start-chat コマンド**を使用して、チャット スクリプトを手動で開始します。
4. 「Timeout expecting: CONNECT」には、さまざまな原因が記述されています。**原因 1：** ローカル モデムが実際にコールを発信していません。 モデムへのリバース Telnet を実行してダイヤルを手動で開始し、モデムがコールを発信できることを確認します。 リモート エンドが応答しない場合は、モデムからコールが発信されていることを確認するため、**ATDT<number>** コマンドを使用してローカル番号に手動で発信し、呼出音を聞きます。 CAS の T1 または E1 と組み込みデジタル モデムを経由した発信コールの場合、トラブルシューティングの大半は他の DDR のトラブルシューティングと同様です。 PRI 回線を経由した組み込みモデムの発信コールの場合でも、同じことがいえます。 この方式によるコール発信でコールが失敗した場合には、コール発信に関与している独自の機能について特別なデバッグが必要になります。他の DDR と同様に、コール試行が要求されたことを確認する必要があります。 これには debug dialer events を使用します。 前述の「*IOS DDR* 」を参照してください。コールを発信するには、その前にモデムをコール用に割り当てておく必要があります。 このプロセスと以降のコールを確認するには、次のデバッグ コマンドを使用します。router# **debug modem** router# **debug modem csm** router# **debug cas** **注:** debug cas コマンドは、AS5200 および AS5300 用に IOS バージョン 12.0(7)T で初めて導入されたものです

。 これよりも前のバージョンの IOS では、システムレベルの設定コマンド service internal と exec コマンド modem-mgmt debug rbs を併用します。デバッグのオン：router#**conf t** Enter configuration commands, one per line. End with CNTL/Z. router(config)#**service internal** router(config)#**^Z** router#**modem-mgmt csm ?** debug-rbs enable rbs debugging no-debug-rbs disable rbs debugging router#**modem-mgmt csm debug-rbs** router# neat msg at slot 0: debug-rbs is on neat msg at slot 0: special debug-rbs is on デバッグのオフ：router#**modem-mgmt csm no-debug-rbs** neat msg at slot 0: debug-rbs is off AS5800 でこれらの情報をデバッグするためには、トランク カードに接続する必要があります。 次の例は、FXS グラウンド スタート用に設定された、CAS T1 経由の通常の発信コールです。Mica Modem(1/0): Rcvd Dial String(5551111)
[Modem receives digits from chat script]

CSM_PROC_IDLE: CSM_EVENT_MODEM_OFFHOOK at slot 1, port 0

CSM_RX_CAS_EVENT_FROM_NEAT:(A003):
EVENT_CHANNEL_LOCK at slot 1 and port 0

CSM_PROC_OC4_DIALING:
CSM_EVENT_DSX0_BCHAN_ASSIGNED at slot 1, port 0

Mica Modem(1/0): Configure(0x1)

Mica Modem(1/0): Configure(0x2)

Mica Modem(1/0): Configure(0x5)

Mica Modem(1/0): Call Setup

neat msg at slot 0: (0/2): Tx RING_GROUND

Mica Modem(1/0): State Transition to Call Setup

neat msg at slot 0: (0/2): Rx TIP_GROUND_NORING
[Telco switch goes OFFHOOK]

CSM_RX_CAS_EVENT_FROM_NEAT:(A003):
EVENT_START_TX_TONE at slot 1 and port 0

CSM_PROC_OC5_WAIT_FOR_CARRIER:
CSM_EVENT_DSX0_START_TX_TONE at slot 1, port 0

neat msg at slot 0: (0/2):
Tx LOOP_CLOSURE [Now the router goes OFFHOOK]

Mica Modem(1/0): Rcvd Tone detected(2)

Mica Modem(1/0): Generate digits:called_party_num=5551111 len=8

Mica Modem(1/0): Rcvd Digits Generated

CSM_PROC_OC5_WAIT_FOR_CARRIER:
CSM_EVENT_ADDR_INFO_COLLECTED at slot 1, port 0

CSM_RX_CAS_EVENT_FROM_NEAT:(A003):
EVENT_CHANNEL_CONNECTED at slot 1 and port 0

CSM_PROC_OC5_WAIT_FOR_CARRIER:
CSM_EVENT_DSX0_CONNECTED at slot 1, port 0

Mica Modem(1/0): Link Initiate

Mica Modem(1/0): State Transition to Connect

```
Mica Modem(1/0): State Transition to Link

Mica Modem(1/0): State Transition to Trainup

Mica Modem(1/0): State Transition to EC Negotiating

Mica Modem(1/0): State Transition to Steady State

Mica Modem(1/0): State Transition to Steady State Speedshifting

Mica Modem(1/0): State Transition to Steady State
```

他のシグナリング タイプの T1 および E1 におけるデバッグも同様です。デバッグでこの時点にまで達した場合は、発信側モデムと応答側モデムの train と接続が完了し、より上位層のプロトコルがネゴシエートを開始できることを示しています。 モデムが発信コール用に正しく割り当てられているにもかかわらず、この時点で接続が失敗する場合は、T1 を調べる必要があります。 **show controller t1/e1 コマンド**を使用して、T1/E1 が機能していることを確認します。 **show controller** の出力の説明については、「シリアル回線のトラブルシューティング」を参照してください。 T1/E1 が正しく機能していない場合は、T1/E1 のトラブルシューティングを行う必要があります。**原因 2：** リモート モデムが応答していません。 通常の電話機を使用してリモート モデムにダイヤルし、動作をテストします。 次の手順を試してください。コールした電話番号が正しいことを確認します。 ハンドセットを使用して着信番号をコールします。手動コールが応答側の非同期モデムに到達できることを確認します。 手動コールが応答側の非同期モデムに到達できる場合は、発信音声コールを許可するように CAS 回線がプロビジョニングされていない可能性があります。手動コールが応答非同期モデムに到達できない場合は、着信側モデムの電話ケーブルを取り換えてから、着信側モデムの回線で通常の電話機を使って試してみます。 通常の電話機でコールが受信できる場合は、着信側モデムに問題がある可能性があります。手動コールが当該回線で通常の電話機に到達しない場合は、着信側施設の別の（確認済みの良好な）回線を使用してみます。正常に接続したら、着信側モデムに接続する電話回線の検査を電話会社に依頼します。長距離コールの場合、発信元で別の（確認済みの良好な）長距離番号を試してみます。 この番号で通話できる場合は、着信側施設または回線が長距離コールを受信するようにプロビジョニングされていない可能性があります。 発信側（CAS）回線が他の長距離番号に到達できない場合は、長距離コールが有効になっていない可能性があります。 別の長距離通話会社で 10-10 コードを試行します。最後に、発信元 CAS 回線から（確認済みの良好な）別のローカル番号を試行します。 それでも接続が失敗する場合は、電話会社に CAS の検査を依頼してください。**原因 3：** ダイヤルしようとしている番号に誤りがあります。 手動でダイヤリングを行い、番号を確認します。 必要であれば設定を訂正します。**原因 4：** モデムの trainup に時間がかかりすぎているか、または TIMEOUT の値が小さすぎます。 **chat-script コマンド**で TIMEOUT 値を大きくしてみます。 TIMEOUT がすでに 60 秒以上の場合は、モデムと、モデムに接続している DTE の間のケーブルに問題がある可能性があります。 trainup が失敗する場合は、回線の問題、またはモデムに互換性がないことが考えられます。個々のモデムの問題の根本的な原因を突き止めるには、reverse telnet を使用して発信元モデムで AT プロンプトに移動します。 可能であれば、リバース Telnet を使用して着信側モデムでも AT プロンプトに移動します。 ATM1 を使用して、モデムがそのスピーカに音声を送信するようにします。これにより、回線で何が起こっているかを両側で聞くことができます。音楽に雑音が混じっているかを確認します。 雑音が混じっている場合は、回線をクリーンアップします。 非同期モデムが trainup していない場合は、番号に発信してスタティックを聞きます。 他の要素が trainup と干渉している可能性があります。 非同期モデムへリバース Telnet を実行してデバッグします。

5. すべてが正しく動作しているものの、CAS 非同期モデム DDR で接続できない場合は、PPP

デバッグを試行します。 チャット スクリプトが正常に終了し、PPP デバッグが障害を示す場合は、『インターネットワーク トラブルシューティング ガイド』の第 17 章にある「PPP のトラブルシューティング」セクションを参照してください。

PRI 非同期モデム DDR

1. PRI 非同期モデム DDR を確認するには、次のコマンドを使用してから、コール発信を試行します。**警告：** *ビジー状態のシステムでデバッグを実行すると、CPU への過負荷またはコンソール バッファのオーバーランが原因でルータがクラッシュすることがあります。*
   `router# debug modem router# debug chat router# debug modem csm router# debug isdn q931 router# debug isdn router# debug ppp negotiate router# debug ppp authenticate`
2. モデム コールの場合、コールを続行するためチャット スクリプトが実行される必要があります。 ダイヤラ マップ ベース DDR の場合、チャット スクリプトはダイヤラ マップ コマンドの **modem-script パラメータ**によって起動されます。 DDR がダイヤラ プロファイル ベースの場合は、TTY 回線で設定されたコマンド script **dialer** によって行われます。 どちらの方法でも、ルータのグローバル コンフィギュレーション内に存在する次のようなチャット スクリプトを利用します。`chat-script callout AT OK atdt\T TIMEOUT 60 CONNECT \c`いずれの場合でも、チャット スクリプトのアクティビティを表示するコマンドは **debug chat** です。 たとえば **dialer map または dialer string コマンド**内のダイヤル文字列（電話番号）が 5551212 であれば、デバッグ出力は次のようになります。`CHAT1: Attempting async line dialer script`

   ```
   CHAT1: Dialing using Modem script: callout & System script: none
   CHAT1: process started
   CHAT1: Asserting DTR
   CHAT1: Chat script callout started
   CHAT1: Sending string: AT
   CHAT1: Expecting string: OK
   CHAT1: Completed match for expect: OK
   CHAT1: Sending string: atdt5551212
   CHAT1: Expecting string: CONNECT
   CHAT1: Completed match for expect: CONNECT
   CHAT1: Chat script callout finished, status = Success
   ```
3. チャット スクリプトが「Sending string」に基づいて予期されるコール（正しい番号）を試行することを確認します。 チャット スクリプトが予期されるコールを発信しない場合は、チャット スクリプトの設定を確認します。 exec プロンプトで **start-chat コマンド**を使用して、チャット スクリプトを手動で開始します。
4. 「Timeout expecting: CONNECT」には、さまざまな原因が記述されています。**原因 1：** ローカル モデムが実際にコールを発信していません。 モデムへのリバース Telnet を実行してダイヤルを手動で開始し、モデムがコールを発信できることを確認します。 リモート エンドが応答しない場合は、モデルからコールが発信されていることを確認するため、**ATDT<number>** コマンドを使用してローカル番号に手動で発信し、呼出音を聞きます。 コールがアウトになったら、ISDN デバッグを有効にします。 BRI で ISDN の障害が最初に疑われる場合は、常に **show isdn status** からの出力をチェックしてください。 ここで注意する重要な点は、レイヤ 1 が *Active* であり、レイヤ 2 が *MULTIPLE_FRAME_ESTABLISHED* の状態にあるということです。 この出力の読み方と修正措置については、『インターネットワーク トラブルシューティング ガイド』の「第 16 章」の「show ISDN ステータスの解釈」を参照してください。ISDN 発信コールの場合、**debug isdn q931** および **debug isdn events** が最適なツールです。 幸い、発信コールのデバッグは着信コールのデバッグと非常によく似ています。 コールが成功すると通常は次のようになります。`*Mar 20 21:07:45.025:`

```
ISDN SE0:23: Event:
Call to 5553759 at 64 Kb/s

*Mar 20 21:07:45.033: ISDN SE0:23: TX ->  SETUP pd = 8
callref = 0x2C
*Mar 20 21:07:45.037:         Bearer Capability i = 0x8890
*Mar 20 21:07:45.041:         Channel ID i = 0x83
*Mar 20 21:07:45.041:         Keypad Facility i = 0x35353533373539
*Mar 20 21:07:45.141: ISDN SE0:23: RX <-  CALL_PROC pd = 8
callref = 0xAC
*Mar 20 21:07:45.145:         Channel ID i = 0x89
*Mar 20 21:07:45.157: ISDN SE0:23: received HOST_PROCEEDING
        Channel ID i = 0x0101
*Mar 20 21:07:45.161:   -------------------
        Channel ID i = 0x89
*Mar 20 21:07:45.313: ISDN SE0:23: RX <-  CONNECT pd = 8
callref = 0xAC
*Mar 20 21:07:45.325: ISDN SE0:23: received HOST_CONNECT
```

CONNECT メッセージは、成功を示す主要なインジケータであることに注意してください。 CONNECT を受信しない場合は、DISCONNECT または RELEASE_COMP（解放完了）メッセージとその後に理由種別コードが表示されます。

```
*Mar 20 22:11:03.212: ISDN SE0:23:
RX <-RELEASE_COMP pd = 8 callref = 0x8F
*Mar 20 22:11:03.216:         Cause i = 0x8295 - Call rejected
```

この理由種別は 2 つのことを示しています。4 バイトまたは 6 バイト値の第 2 バイトは、エンドツーエンドのコール パス内のどこから DISCONNECT または RELEASE_COMP を受信したかを示しています。 これを問題の特定に役立てることができます。第 3 バイトおよび第 4 バイトは、障害の実際の理由を示しています。 それぞれの値の意味については、「表 9」を参照してください。**原因 2：** リモート モデムが応答していません。 通常の電話機を使用してリモート モデムにダイヤルし、動作をテストします。 次の手順を試してください。コールした電話番号が正しいことを確認します。 ハンドセットを使用して着信番号をコールします。手動コールが応答側の非同期モデムに到達できることを確認します。 手動コールが応答側の非同期モデムに到達できる場合は、発信音声コールを許可するように BRI 回線がプロビジョニングされていない可能性があります。手動コールが応答非同期モデムに到達できない場合は、着信側モデムの電話ケーブルを取り換えてから、着信側モデムの回線で通常の電話機を使って試してみます。 通常の電話機でコールが受信できる場合は、着信側モデムに問題がある可能性があります。手動コールが当該回線で通常の電話機に到達しない場合は、着信側施設の別の（確認済みの良好な）回線を使用してみます。 正常に接続したら、着信側モデムに接続する電話回線の検査を電話会社に依頼します。長距離コールの場合、発信元で別の（確認済みの良好な）長距離番号を試してみます。 この番号で通話できる場合は、着信側施設または回線が長距離コールを受信するようにプロビジョニングされていない可能性があります。 発信側（BRI）回線が他の長距離番号に到達できない場合は、長距離コールが有効になっていない可能性があります。 別の長距離通話会社で 1010 コードを試行します。最後に、発信元 BRI 回線から（確認済みの良好な）別のローカル番号を試行します。 それでも接続が失敗する場合は、電話会社に BRI の検査を依頼してください。**原因 3：** ダイヤルしようとしている番号に誤りがあります。 手動でダイヤリングを行い、番号を確認します。 必要であれば設定を訂正します。**原因 4：** モデムの trainup に時間がかかりすぎているか、または TIMEOUT の値が小さすぎます。 chat-script コマンドで TIMEOUT 値を大きくしてみます。 TIMEOUT がすでに 60 秒以上の場合は、モデムと、モデムに接続している DTE の間にケーブルの問題がある可能性があります。 trainup が失敗する場合は、回線の問題、またはモデムに互換性がないことが考えられます。個々のモデムの問題の根本的な原因を突き止めるには、reverse telnet を使用して発信元モデムで AT プロンプトに移動します。 可能であれば、リバース Telnet を使用して着信側モデムでも AT プロンプトに移動

します。 ATM1 を使用して、モデムがそのスピーカに音声を送信するようにします。これにより、回線で何が起こっているかを両側で聞くことができます。音楽に雑音が混じっているかを確認します。 雑音が混じっている場合は、回線をクリーンアップします。 非同期モデムが trainup していない場合は、番号に発信してスタティックを聞きます。 他の要素が trainup と干渉している可能性があります。 非同期モデムへリバース Telnet を実行してデバッグします。

5. すべてが正しく動作しているものの、BRI 非同期モデム DDR で接続できない場合は、PPP デバッグを試行します。 チャット スクリプトが正常に終了し、PPP デバッグが障害を示す場合は、『インターネットワーク トラブルシューティング ガイド』の第 17 章にある「PPP のトラブルシューティング」セクションを参照してください。

BRI 非同期モデム DDR

この機能は、Cisco IOS ソフトウェア リリース 12.0(3)T 以降が稼働する Cisco 3640 プラットフォームでのみ動作します。 この機能には、BRI ネットワーク モジュールの最新のハードウェア リビジョンが必要です。 これは WAN インターフェイス カード（WIC）では機能しません。

1. show modem コマンドで国番号が正しいことを確認します。 次のコマンドを使用してから、コール発信を試行します。**警告：** *ビジー状態のシステムでデバッグを実行すると、CPU への過負荷またはコンソール バッファのオーバーランが原因でルータがクラッシュすることがあります。*`router# debug modem router# debug chat router# debug modem csm router# debug isdn q931 router# debug bri router# debug ppp negotiate router# debug ppp authenticate`
2. モデム コールの場合、コールを続行するためチャット スクリプトが実行される必要があります。 ダイヤラ マップ ベース DDR の場合、チャット スクリプトはダイヤラ マップ コマンドの **modem-script パラメータ**によって起動されます。 DDR がダイヤラ プロファイル ベースの場合は、TTY 回線で設定されたコマンド script **dialer** によって行われます。 どちらを使用する場合も、ルータのグローバル コンフィギュレーション内に存在する次のようなチャット スクリプトを利用します。`chat-script callout AT OK atdt\T TIMEOUT 60 CONNECT \c` いずれの場合でも、チャット スクリプトのアクティビティを表示するコマンドは **debug chat** です。 たとえば **dialer map または dialer string コマンド**内のダイヤル文字列（電話番号）が 5551212 であれば、デバッグ出力は次のようになります。`CHAT1: Attempting async line dialer script`

```
CHAT1: Dialing using Modem script: callout & System script: none
CHAT1: process started
CHAT1: Asserting DTR
CHAT1: Chat script callout started
CHAT1: Sending string: AT
CHAT1: Expecting string: OK
CHAT1: Completed match for expect: OK
CHAT1: Sending string: atdt5551212
CHAT1: Expecting string: CONNECT
CHAT1: Completed match for expect: CONNECT
CHAT1: Chat script callout finished, status = Success
```

3. チャット スクリプトが「Sending string」に基づいて予期されるコール（正しい番号）を試行することを確認します。 チャット スクリプトが予期されるコールを発信しない場合は、チャット スクリプトの設定を確認します。 exec プロンプトで **start-chat コマンド**を使用して、チャット スクリプトを手動で開始します。
4. 「Timeout expecting: CONNECT」には、さまざまな原因が記述されています。**原因 1：** ローカル モデムが実際にコールを発信していません。 モデムへのリバース Telnet を実行してダイヤルを手動で開始し、モデムがコールを発信できることを確認します。 リモート エン

ドが応答しない場合は、モデルからコールが発信されていることを確認するため、**ATDT<number>** コマンドを使用してローカル番号に手動で発信し、呼出音を聞きます。 コールがアウトになったら、ISDN デバッグを有効にします。 BRI で ISDN の障害が最初に疑われる場合は、常に **show isdn status** からの出力をチェックしてください。 ここで注意する重要な点は、レイヤ 1 が **Active** であり、レイヤ 2 が **MULTIPLE_FRAME_ESTABLISHED** の状態にあるということです。 この出力の読み方と修正措置については、『インターネットワーク トラブルシューティング ガイド』の「第 16 章」の「show ISDN ステータスの解釈」を参照してください。ISDN 発信コールの場合、**debug isdn q931** および **debug isdn events** が最適なツールです。 幸い、発信コールのデバッグは着信コールのデバッグと非常によく似ています。 コールが成功すると通常は次のようになります。

```
*Mar 20 21:07:45.025: ISDN BR0: Event:
Call to 5553759 at 64 Kb/s

*Mar 20 21:07:45.033: ISDN BR0: TX ->  SETUP pd = 8
callref = 0x2C
*Mar 20 21:07:45.037:         Bearer Capability i = 0x8890
*Mar 20 21:07:45.041:         Channel ID i = 0x83
*Mar 20 21:07:45.041:         Keypad Facility i = 0x35353533373539
*Mar 20 21:07:45.141: ISDN BR0: RX <-  CALL_PROC pd = 8
callref = 0xAC
*Mar 20 21:07:45.145:         Channel ID i = 0x89
*Mar 20 21:07:45.157: ISDN BR0: received HOST_PROCEEDING
        Channel ID i = 0x0101
*Mar 20 21:07:45.161:   -------------------
        Channel ID i = 0x89
*Mar 20 21:07:45.313: ISDN BR0: RX <-  CONNECT pd = 8
callref = 0xAC
*Mar 20 21:07:45.325: ISDN BR0: received HOST_CONNECT
```

CONNECT メッセージは、成功を示す主要なインジケータであることに注意してください。 CONNECT を受信しない場合は、DISCONNECT または RELEASE_COMP（解放完了）メッセージとその後に理由種別コードが表示されます。

```
*Mar 20 22:11:03.212: ISDN BR0: RX <-  RELEASE_COMP pd = 8
callref = 0x8F
*Mar 20 22:11:03.216:         Cause i = 0x8295 - Call rejected
```

この理由種別は 2 つのことを示しています。4 バイトまたは 6 バイト値の第 2 バイトは、エンドツーエンドのコール パス内のどこから DISCONNECT または RELEASE_COMP を受信したかを示しています。 これを問題の特定に役立てることができます。第 3 バイトおよび第 4 バイトは、障害の実際の理由を示しています。 それぞれの値の意味については、「表 9」を参照してください。**原因 2：** リモート モデムが応答していません。 通常の電話機を使用してリモート モデムにダイヤルし、動作をテストします。 次の手順を試してください。コールした電話番号が正しいことを確認します。 ハンドセットを使用して着信番号をコールします。手動コールが応答側の非同期モデムに到達できることを確認します。 手動コールが応答側の非同期モデムに到達できる場合は、発信音声コールを許可するように BRI 回線がプロビジョニングされていない可能性があります。手動コールが応答非同期モデムに到達できない場合は、着信側モデムの電話ケーブルを取り換えてから、着信側モデムの回線で通常の電話機を使って試してみます。 通常の電話機でコールが受信できる場合は、着信側モデムに問題がある可能性があります。手動コールが当該回線で通常の電話機に到達しない場合は、着信側施設の別の（確認済みの良好な）回線を使用してみます。 正常に接続したら、着信側モデムに接続する電話回線の検査を電話会社に依頼します。長距離コールの場合、発信元で別の（確認済みの良好な）長距離番号を試してみます。 この番号で通話できる場合は、着信側施設または回線が長距離コールを受信するようにプロビジョニングされていない可能性があります。 発信側（BRI）回線が他の長距離番号に到達できない場合は、長距離コールが有効になっていない可能性があります。 別の長距離通話会社で 10-10 コード

を試行します。最後に、発信元 BRI 回線から（確認済みの良好な）別のローカル番号を試行します。 それでも接続が失敗する場合は、電話会社に BRI の検査を依頼してください。**原因 3**： ダイヤルしようとしている番号に誤りがあります。 手動でダイヤリングを行い、番号を確認します。 必要であれば設定を訂正します。**原因 4**： モデムの trainup に時間がかかりすぎているか、または TIMEOUT の値が小さすぎます。 chat-script コマンドで TIMEOUT 値を大きくしてみます。 TIMEOUT がすでに 60 秒以上の場合は、モデムと、モデムに接続している DTE の間にケーブルの問題がある可能性があります。 trainup が失敗する場合は、回線の問題、またはモデムに互換性がないことが考えられます。個々のモデムの問題の根本的な原因を突き止めるには、reverse telnet を使用して発信元モデムで AT プロンプトに移動します。 可能であれば、リバース Telnet を使用して着信側モデムでも AT プロンプトに移動します。 ATM1 を使用して、モデムがそのスピーカに音声を送信するようにします。これにより、回線で何が起こっているかを両側で聞くことができます。音楽に雑音が混じっているかを確認します。 雑音が混じっている場合は、回線をクリーンアップします。 非同期モデムが trainup していない場合は、番号に発信してスタティックを聞きます。 他の要素が trainup と干渉している可能性があります。 非同期モデムへリバース Telnet を実行してデバッグします。

5. すべてが正しく動作しているものの、BRI 非同期モデム DDR で接続できない場合は、PPP デバッグを試行します。 チャット スクリプトが正常に終了し、PPP デバッグが障害を示す場合は、『インターネットワーク トラブルシューティング ガイド』の第 17 章にある「PPP のトラブルシューティング」セクションを参照してください。

PRI ISDN DDR

1. PRI で ISDN の障害が最初に疑われる場合は、常に **show isdn status** からの出力をチェックしてください。 ここで注意する重要な点は、レイヤ 1 が Active であり、レイヤ 2 が *MULTIPLE_FRAME_ESTABLISHED* の状態にあるということです。 この出力の読み方と修正措置については、『インターネットワーク トラブルシューティング ガイド』の「第 16 章」の「show ISDN ステータスの解釈」を参照してください。ISDN 発信コールの場合、**debug isdn q931** および **debug isdn events** が最適なツールです。 幸い、発信コールのデバッグは着信コールのデバッグと非常によく似ています。 コールが成功すると通常は次のようになります。
```
*Mar 20 21:07:45.025: ISDN SE0:23: Event:
Call to 5553759 at 64 Kb/s

*Mar 20 21:07:45.033: ISDN SE0:23: TX ->  SETUP pd = 8
callref = 0x2C
*Mar 20 21:07:45.037:         Bearer Capability i = 0x8890
*Mar 20 21:07:45.041:         Channel ID i = 0x83
*Mar 20 21:07:45.041:         Keypad Facility i = 0x35353533373539
*Mar 20 21:07:45.141: ISDN SE0:23: RX <-  CALL_PROC pd = 8
callref = 0xAC
*Mar 20 21:07:45.145:         Channel ID i = 0x89
*Mar 20 21:07:45.157: ISDN SE0:23: received HOST_PROCEEDING
        Channel ID i = 0x0101
*Mar 20 21:07:45.161:   -------------------
        Channel ID i = 0x89
*Mar 20 21:07:45.313: ISDN SE0:23: RX <-  CONNECT pd = 8
callref = 0xAC
*Mar 20 21:07:45.325: ISDN SE0:23: received HOST_CONNECT
```
CONNECT メッセージは、成功を示す主要なインジケータであることに注意してください。 CONNECT を受信しない場合は、DISCONNECT または RELEASE_COMP（解放完了）メッセージとその後に理由種別コードが表示されます。
```
*Mar 20 22:11:03.212: ISDN SE0:23: RX <-  RELEASE_COMP pd = 8
```

```
callref = 0x8F
*Mar 20 22:11:03.216:         Cause i = 0x8295 - Call rejected
```

この理由種別は 2 つのことを示しています。4 バイトまたは 6 バイト値の第 2 バイトは、エンドツーエンドのコール パス内のどこから DISCONNECT または RELEASE_COMP を受信したかを示しています。 これを問題の特定に役立てることができます。第 3 バイトおよび第 4 バイトは、障害の実際の理由を示しています。 それぞれの値の意味については、「表 9」を参照してください。**注:** 次の出力は、上位層プロトコルの障害を示しています。`Cause i = 0x8090 - Normal call clearing`PPP 認証の失敗が典型的な理由です。 接続障害が必ず ISDN の問題であると見なす前に、**debug ppp negotiation** および **debug ppp authentication** をオンにします。

2. ISDN CONNECT メッセージが表示され、PPP デバッグが障害を示す場合は、『インターネットワーク トラブルシューティング ガイド』の「第 17 章」にある「PPP のトラブルシューティング」セクションを参照してください。

BRI ISDN DDR

1. BRI で ISDN の障害が最初に疑われる場合は、常に show isdn status からの出力をチェックしてください。 ここで注意する重要な点は、レイヤ 1 が Active であり、レイヤ 2 が MULTIPLE_FRAME_ESTABLISHED の状態にあるということです。 この出力の読み方と修正措置については、『インターネットワーク トラブルシューティング ガイド』の「第 16 章」の「show ISDN ステータスの解釈」を参照してください。ISDN 発信コールの場合、**debug isdn q931** および **debug isdn events** が最適なツールです。 幸い、発信コールのデバッグは着信コールのデバッグと非常によく似ています。 コールが成功すると通常は次のようになります。`*Mar 20 21:07:45.025: ISDN BR0: Event: Call to 5553759 at 64 Kb/s`

```
*Mar 20 21:07:45.033: ISDN BR0: TX ->  SETUP pd = 8  callref = 0x2C
*Mar 20 21:07:45.037:         Bearer Capability i = 0x8890
*Mar 20 21:07:45.041:         Channel ID i = 0x83
*Mar 20 21:07:45.041:         Keypad Facility i = 0x35353533373539
*Mar 20 21:07:45.141: ISDN BR0: RX <-  CALL_PROC pd = 8  callref = 0xAC
*Mar 20 21:07:45.145:         Channel ID i = 0x89
*Mar 20 21:07:45.157: ISDN BR0: received HOST_PROCEEDING
        Channel ID i = 0x0101
*Mar 20 21:07:45.161:   -------------------
        Channel ID i = 0x89
*Mar 20 21:07:45.313: ISDN BR0: RX <-  CONNECT pd = 8  callref = 0xAC
*Mar 20 21:07:45.325: ISDN BR0: received HOST_CONNECT
```

CONNECT メッセージは、成功を示す主要なインジケータであることに注意してください。 CONNECT を受信しない場合は、DISCONNECT または RELEASE_COMP（解放完了）メッセージとその後に理由種別コードが表示されます。`*Mar 20 22:11:03.212: ISDN BR0: RX <-  RELEASE_COMP pd = 8`

```
callref = 0x8F
*Mar 20 22:11:03.216:         Cause i = 0x8295 - Call rejected
```

この理由種別は 2 つのことを示しています。4 バイトまたは 6 バイト値の第 2 バイトは、エンドツーエンドのコール パス内のどこから DISCONNECT または RELEASE_COMP を受信したかを示しています。 これを問題の特定に役立てることができます。第 3 バイトおよび第 4 バイトは、障害の実際の理由を示しています。 それぞれの値の意味については、「表 9」を参照してください。**注:** 次の出力は、上位層プロトコルの障害を示しています。`Cause i = 0x8090 - Normal call clearing`PPP 認証の失敗が典型的な理由です。 接続障害が必ず ISDN の問題であると見なす前に、**debug ppp negotiation** および **debug ppp authentication** をオンにします。

2. ISDN CONNECT メッセージが表示され、PPP デバッグが障害を示す場合は、『インターネ

ットワーク トラブルシューティング ガイド』の 「第 17 章」にある「PPP のトラブルシューティング」セクションを参照してください。

関連情報

- Cisco IOS ダイヤル サービス クイック コンフィギュレーション ガイド
- Cisco IOS ダイヤル サービス コンフィギュレーション ガイド ネットワーク サービス
- Cisco IOS ダイヤル サービス コンフィギュレーション ガイド Terminal Services
- Cisco IOS ダイヤル サービス コマンド リファレンス
- ダイヤル ケース スタディの概要
- テクノロジー ページへのアクセス
- テクニカルサポートとドキュメント - Cisco Systems