

# VIC-2DID の設定とトラブルシューティング

## 内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[表記法](#)

[背景説明](#)

[コール スタート シグナリング](#)

[イミディエート スタート](#)

[ウイंक スタート](#)

[ディレイ ダイアル](#)

[コール監視](#)

[応答監視](#)

[切断監視](#)

[失敗したコールの処理](#)

[ダイヤルイン方式の設定](#)

[トラブルシュート](#)

[関連情報](#)

## 概要

このドキュメントでは、Foreign Exchange Station ( FXS ) - 2DID カードの基本設定の実装方法と、シグナリングの使用方法を説明します。ハードウェアおよび Cisco IOS® ソフトウェア サポートの詳細については、『Understanding 2 Port Direct Inward Dial (2 DID) Voice Interface Cards』を参照してください。

## 前提条件

### 要件

このドキュメントに特有の要件はありません。

### 使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- Cisco VG200 ゲートウェイ
- Cisco IOS ソフトウェア リリース 12.2(8)T

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ました。このド

キュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。対象のネットワークが実稼働中である場合には、どのようなコマンドについても、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。

## [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。

## [背景説明](#)

ダイヤルイン方式（DID）は、発信者が、オペレータや自動化されたコールアテンダントの支援を得ることなく、構内交換機（PBX）またはパケット音声システム（たとえば Cisco CallManager および IOS ルータ/ゲートウェイなど）の内線に直接ダイヤルできる、電話会社から提供されるサービスです。このサービスは、電話番号の末尾 3 ～ 5 桁のみを PBX またはルータ/ゲートウェイに転送する DID トランクを使用します。たとえば、ある企業が電話の内線番号 555-1000 から 555-1999 までを所有しており、発信者が 555-1234 をダイヤルした場合、ローカルの Central Office（CO）は 234 を PBX またはパケット音声システムに転送します。PBX またはパケット音声システムは内線番号 234 を鳴らします。また、DID トランクは最大 50 の内線番号を処理できるため、比較的少ないトランクで多数の内線番号を処理できます。

## [コール スタート シグナリング](#)

DID はディジットを送信する前に、DID トランクと PBX 間のハンドシェイク プロトコルを要求します。シグナリングは、ウイंक スタート、ディレイダイヤル、またはイミディエート スタートです。これらのシグナリングタイプは Ear and Mouth（E&M）シグナリングで使用されるものと同様です。

### [イミディエート スタート](#)

イミディエート スタートは最もシンプルなプロトコルです。発信側はオフフックにし、応答を待つことなくディジットの点滅を開始することで、回線を捕捉します。標準に従えば、イミディエート スタートで使用されるアドレスシグナリングはダイヤル点滅です。

### [ウイंक スタート](#)

ウイंक スタートでは、発信側はオフフックにし、パルス送信を開始する前に、他方からの確認応答を待つことで、回線を捕捉します。確認応答は 140 ～ 290 ミリ秒間の極性反転（オフフック）で、これもウイंकと呼ばれます。ウイंकは、着信捕捉シグナルの受信後 100 ミリ秒以内は発生しません。シグナリング機能に加えて、ウイंक スタートは故障トランクを識別する整合性チェックの役割も果たし、ネットワークが発信者にリオーダー音を送信できるようにします。

### [ディレイダイヤル](#)

ディレイダイヤルモードでは、発信側が回線を捕捉し（オフフックになり）、約 200 ミリ秒間待機し、相手側がオンフック（バッテリ通常）であるかどうかを確認します。その場合、ダイヤルディジットを出力します。相手がオフフック（バッテリ反転）である場合には、オンフック（バッテリ通常）になるまで待機し、それからダイヤルディジットを出力します。

# コール監視

## 応答監視

応答監視は極性反転シグナルです。応答監視は、コールがステーションによって応答された場合、あるいは録音メッセージまたは自動音声応答装置 (IVR) にルーティングされた場合に、返されます。応答監視を返さない条件は、内線がビジーのとき、または番号が使用できないときです。

注：応答監視が電話会社の機器で正しく設定されていない場合は、単方向音声が発生する可能性があります。

## 切断監視

発信側が終了する前に呼び出されたステーションが切断した場合、バッテリーが通常に戻り、着信側でオンフックを待機します。トランクが切断した後に呼び出されたステーションが切断しなかった場合、通常のバッテリー電圧は DID ポートで復元され、トランクとステーション間の接続をドロップします。

## 失敗したコールの処理

コールが未割り当て番号または永続的に制限されたステーションに着信すると、発信者はリオーダー音を受け取ります。呼び出されたユーザがオフフックの場合、発信者はビジー トーンを受け取ります。

# ダイヤルイン方式の設定

これらの DID に関連付けられたコマンドライン インターフェイス (CLI) コマンドは、音声ポートのハードウェアが DID 機能をサポートしている場合のみ有効です。また、DID と E&M ポートは同様の機能を備えているため、E&M の音声ポートで使用されるすべての CLI コマンドは DID ポートでも使用できます。デフォルトの動作モードは DID です。

FXS-DID 音声ポートを設定する：

```
voice-port x/y
signal did    wink-start
delay-start
immediate-start
did-digit-length <Digit length>
```

通常の FXS にするために FXS-DID を無効にする：

```
voice-port x/y
no signal did
```

注：発信コールは、DID (設定済み) 音声ポートでは行えません。ハードウェアがこれらのポートでの呼び出し音配置と発信をサポートできても、このソフトウェアを使用して無効化されます。

# トラブルシュート

DID ラインの特性を通信事業者が確認することは重要であり、DID カードの設定はこの確認された情報に基づきます。最も一般的な問題は、ダイヤルプランの誤設定 ( 何桁が CO から提供されたか )、誤ったコールシグナリングや回線極性です ( 通信機器は極性に敏感なため、チップアンドリング接続を逆にしなければならない場合があります )。

これは、番号 4609 をコールするウインク DID トランクからの debug vpm all コマンドの出力です。

```
*Mar 5 00:54:22.783: htsp_dsp_message: SEND/RESP_SIG_STATUS: state=0xC
timestamp=13671 systime=34886280
*Mar 5 00:54:22.783: htsp_process_event: [1/0/0, DID_ONHOOK, E_DSP_SIG_1100]
did_onhook_offhook htsp_setup_ind
*Mar 5 00:54:22.787: [1/0/0] get_local_station_id calling num= calling name=
calling time=00/00 00:00
*Mar 5 00:54:22.791: dsp_digit_collect_on: [1/0/0] packet_len=20 channel_id=128
packet_id=35 min_inter_delay=240 max_inter_delay=9760
mim_make_time=10 max_make_time=100 min_brake_time=10
max_brake_time=100
*Mar 5 00:54:22.791: dsp_soutput: [1/0/0]
*Mar 5 00:54:22.795: dsp_digit_collect_on: [1/0/0] packet_len=20 channel_id=128
packet_id=35 min_inter_delay=240 max_inter_delay=9760
mim_make_time=10 max_make_time=100 min_brake_time=10
max_brake_time=100
*Mar 5 00:54:22.795: dsp_soutput: [1/0/0]
*Mar 5 00:54:22.795: dsp_digit_collect_on: [1/0/0] packet_len=20 channel_id=128
packet_id=35 min_inter_delay=240 max_inter_delay=9760
mim_make_time=10 max_make_time=100 min_brake_time=10
max_brake_time=100
*Mar 5 00:54:22.795: dsp_soutput: [1/0/0]
*Mar 5 00:54:22.795: htsp_process_event: [1/0/0, DID_WAIT_SETUP_ACK,
E_HTSP_SETUP_ACK]did_wait_setup_ack_get_ack
*Mar 5 00:54:22.795: dsp_digit_collect_off: [1/0/0] packet_len=8 channel_id=128
packet_id=36
*Mar 5 00:54:22.795: dsp_soutput: [1/0/0]
*Mar 5 00:54:22.799: htsp_timer2 - 88 msec
*Mar 5 00:54:22.799: htsp_dsp_message: SEND/RESP_SIG_STATUS: state=0xC
timestamp=13685 systime=34886282
*Mar 5 00:54:22.799: htsp_process_event: [1/0/0, DID_WAIT_SETUP_ACK,
E_DSP_SIG_1100]did_wait_setup_ack_offhook
*Mar 5 00:54:22.799: did_stop_timer
*Mar 5 00:54:22.799: htsp_timer_stop
*Mar 5 00:54:22.887: htsp_process_event: [1/0/0, DID_WAIT_SETUP_ACK,
E_HTSP_EVENT_TIMER2] did_wait_prewink_timer
*Mar 5 00:54:22.887: dsp_digit_collect_off: [1/0/0] packet_len=8 channel_id=128
packet_id=36
*Mar 5 00:54:22.887: dsp_soutput: [1/0/0] did_offhook
*Mar 5 00:54:22.887: [1/0/0] set signal state = 0x6 timestamp = 0
*Mar 5 00:54:22.887: dsp_set_sig_state: [1/0/0] packet_len=12 channel_id=128
packet_id=39 state=0x6 timestamp=0x0
*Mar 5 00:54:22.887: dsp_soutput: [1/0/0] did_onhook
*Mar 5 00:54:22.887: [1/0/0] set signal state = 0x4 timestamp = 200
*Mar 5 00:54:22.887: dsp_set_sig_state: [1/0/0] packet_len=12 channel_id=128
packet_id=39 state=0x4 timestamp=0xC8
*Mar 5 00:54:22.887: dsp_soutput: [1/0/0]
*Mar 5 00:54:22.891: dsp_digit_collect_on: [1/0/0] packet_len=20 channel_id=128
packet_id=35 min_inter_delay=240 max_inter_delay=9760
mim_make_time=10 max_make_time=100 min_brake_time=10
max_brake_time=100
*Mar 5 00:54:22.891: dsp_soutput: [1/0/0]
*Mar 5 00:54:23.879: htsp_digit_ready(1/0/0): digit = 4
*Mar 5 00:54:24.983: htsp_digit_ready(1/0/0): digit = 6
```

```
*Mar 5 00:54:26.483: htsp_digit_ready(1/0/0): digit = 0
*Mar 5 00:54:27.891: htsp_digit_ready(1/0/0): digit = 9
*Mar 5 00:54:27.891: dsp_digit_collect_off: [1/0/0] packet_len=8 channel_id=128
packet_id=36
*Mar 5 00:54:27.891: dsp_soutput: [1/0/0]
*Mar 5 00:54:27.895: htsp_dsp_message: SEND/RESP_SIG_STATUS: state=0xC
timestamp=18781 systime=34886792
*Mar 5 00:54:27.895: htsp_process_event: [1/0/0, DID_OFFHOOK, E_HTSP_PROCEEDING]
*Mar 5 00:54:27.895: htsp_process_event: [1/0/0, DID_OFFHOOK, E_DSP_SIG_1100]
did_offhook_offhook
*Mar 5 00:54:27.895: did_stop_timer
*Mar 5 00:54:27.895: htsp_timer_stop wrong offhook eventhtsp_alert_notify
*Mar 5 00:54:32.415: htsp_process_event: [1/0/0, DID_OFFHOOK,
E_HTSP_VOICE_CUT_THROUGH] htsp_connect: no_offhook 0
*Mar 5 00:54:32.419: htsp_process_event: [1/0/0, DID_OFFHOOK, E_HTSP_CONNECT]
did_offhook_connect
*Mar 5 00:54:32.419: htsp_timer2 - 40 msec did_offhook
*Mar 5 00:54:32.419: [1/0/0] set signal state = 0x6 timestamp = 250
*Mar 5 00:54:32.419: dsp_set_sig_state: [1/0/0] packet_len=12 channel_id=128
packet_id=39 state=0x6 timestamp=0xFA
*Mar 5 00:54:32.419: dsp_soutput: [1/0/0]
*Mar 5 00:54:32.459: htsp_process_event: [1/0/0, DID_CONNECT_MIN,
E_HTSP_EVENT_TIMER2]
*Mar 5 00:55:01.659: htsp_dsp_message: SEND/RESP_SIG_STATUS: state=0x4
timestamp=52547 systime=34890168
*Mar 5 00:55:01.659: htsp_process_event: [1/0/0, DID_CONNECT,
E_DSP_SIG_0100]did_offhook_onhook
```

## [関連情報](#)

- [音声に関する技術サポート](#)
- [音声とユニファイド コミュニケーションに関する製品サポート](#)
- [Cisco IP Telephony のトラブルシューティング](#)
- [テクニカルサポート - Cisco Systems](#)