

# VIC-2DID 的配置与故障排除

## 目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[规则](#)

[背景信息](#)

[呼叫启动信令](#)

[立即启动](#)

[闪烁启动](#)

[延迟拨号](#)

[呼叫管理](#)

[应答监督](#)

[断开监督](#)

[失败呼叫的处理](#)

[配置直接拨入](#)

[故障排除](#)

[相关信息](#)

## [简介](#)

本文提供细节关于怎样实现局外交换站的基本配置(FXS)-2DID卡，并且如何使用发信号。关于硬件和Cisco IOS软件支持的更多信息，参考[了解2个波尔特直接拨入\(2个DID\)语音接口卡](#)。

## [先决条件](#)

### [要求](#)

本文档没有任何特定的要求。

### [使用的组件](#)

本文档中的信息基于以下软件和硬件版本：

- Cisco VG200网关
- Cisco IOS软件版本12.2(8)T

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

## 规则

有关文档规则的详细信息，请参阅 [Cisco 技术提示规则](#)。

## 背景信息

直接拨入(DID)是使呼叫方拨号直接地到内部交换机(PBX)或语音信息包系统的一分机电话公司提供的服务(例如，Cisco CallManager和IOS路由器/网关)，不用操作员或自动化呼叫话务员的协助。此服务利用转发电话号码只最后三个到五个位到PBX或路由器/网关的DID的中继。例如，当公司有电话分机555-1000对555-1999和呼叫方拨号555-1234时，本地中心局转发234到PBX或语音信息包系统。PBX或语音信息包系统环分机234。并且，因为DID的中继能服务50扩展，服务很大数量的扩展用相对小量的中继是可能的。

## 呼叫起动信令

DID在位前发射要求在DID的中继和PBX之间的握手协议。信令是WINK启动、延迟拨号或者立即启动。这些信令类型类似于用于耳和嘴的那些(E&M)信令。

### 立即启动

立即启动是最简单协议。始发端通过去占用线路摘机并且开始搏动位，无需等待答复。根据标准，地址信令与立即启动一起使用是拨号脉冲。

### 闪烁启动

使用WINK启动，始发端通过去占用线路摘机，并且在启动前，局间发送脉冲等待从另一端的一确认。确认是极性反转(摘机) 140到290毫秒的持续时间，也呼叫闪烁。闪烁应该发生没有早于100毫秒，在流入捕获信号的收据后。除信令功能之外，WINK启动担当识别一发生故障的中继的完整性检查并且允许网络发送交换机忙音对主叫方。

### 延迟拨号

在延迟拨号模式，始发端占用线路(去摘机)，等待大约200毫秒和检查发现远端是否挂机(电池正常)。如果那样，它输出拨号数位。如果远端摘机(被倒转的电池)，等待，直到是挂机(电池正常)，然后输出了拨号数位。

## 呼叫管理

### 应答监督

应答监督是电池反转信号。应答监督返回，当呼叫由站点时应答，或者路由对一已录制通告或交互语音应答(IVR)。不返回的应答监督条件是，当呼叫的分机忙碌时或编号不可用。

**注意：** 如果应答监督在电信公司设备没有正确地配置，单向语音可能发生。

### 断开监督

当被叫站在始发端前时断开，电池被倒转对正常并且等待挂机在流入末端。如果被叫站未断开，在中继断开，正常电池电压在DID的端口恢复并且切连接在中继和站点之间后。

## 失败呼叫的处理

当呼叫完成到未分配的号码或一个永久限制站点时，呼叫方接收交换机忙音。当呼叫的用户摘机时，呼叫方接收忙音。

## 配置直接拨入

这些DID的相关的命令行界面(CLI)命令只有效，如果语音端口硬件支持DID的功能。另外，因为DID和E&M端口有相似的功能，用于E&M语音端口的所有CLI命令为DID的端口也是可用的。默认操作模式是DID。

配置FXS-DID语音端口：

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

禁用FXS-DID变为正常FXS：

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

**注意：**呼出呼叫在DID (已配置的)语音端口不可能被做。即使硬件可以支持发出环和进行的呼出呼叫在这些端口，使用软件，禁用。

## 故障排除

非常重要是DID的线路的特性检查与Telco，并且DID的卡的配置根据此验证的信息。最常见的问题是拨号计划(多少个位的不正确的配置由CO传送)，不正确呼叫信令和线路极性(电信公司设备是敏感的极性，因此Tip AND Ring连接可能需要被倒转)。

这是输出debug vpm all命令从呼叫第4609的闪烁DID的中继。

```
*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  
min\_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  
min\_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,

## 相关信息

- [语音技术支持](#)
- [语音和统一通信产品支持](#)
- [Cisco IP 电话故障排除](#)
- [技术支持 - Cisco Systems](#)