Troubleshoot and Monitor Analog Ports

Introduction

This document describes Foreign Exchange Station (FXS) and Foreign Exchange Office (FXO) ports and the mechanisms behind how they function. More specifically, it covers how calls are setup and torn down between the two ports. In addition to this, it discusses the different configuration components of the ports and how to troubleshoot them.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Analog Interface Basics

Components Used

The information in this document is based on these hardware and software versions:

- ISR4451-X/K9
- NIM-2FXSP
- NIM-2FXO
- IOS-XE Version 16.8.2

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.
FXS/FXO Fundamentals

Port Fundamentals

FXO/FXS is the typical analog circuit used to deliver a single analog phone line to your house.

An FXO port (or device) is the port that connects to the circuit, which has the ability to go on/off hook, and transmit digits when off-hook to initiate a call. It employs a relay that when the port is off-hook the circuit is closed, and when the port is considered on-hook the circuit is open. It interconnects to an FXS on the other side.

Same as FXO ports terminate lines from your carrier, you can also think of these as your analog handset, or Fax Machine/Modem.

An FXS port is the device that connects to the circuit and provide dial tone and ringing voltage to an FXO device. An FXS port connects a gateway to equipment such as telephones, fax machines, and modems. An FXS port utilizes only 2 wires (Tip and Ring) for both the signaling and audio path on a given call. This two wire pair, it is able to supply ring, voltage, and dial tone to the station.

Basic Call Flow

To better understand how to troubleshoot these two pots types, you first need to look at how a call sets up on them. This section shows the process of analog call from the time both endpoints are on hook, till the point at which there is two way audio.

As with all calls, the FXS port starts in a On-hook state while the two endpoints are not in use:

![Diagram of Basic Call Flow]

When one of the phones go Off-hook, the circuit gets closed and dial tone is provided by the FXS port to the FXO device:
Once the device that has initiated the call is Off-hook, it starts to dial the number by either by Pulses or Tones:

Once the number has been dialed, the device that handles this call routes it accordingly. Once the call has been routed, while the far end device is alerted of the call that it received, the device the call originates from is played Ring Back Tone:

Once the far end device has picked the call up, its circuit is closed as well and it at this point that the call is connected with two way audio:
The previous example, is a basic flow of what happens from the start to the stop of a call. However, there is more that goes on behind the scenes for that FXS port to signal to the phone about each of its call states. The next section covers, the two most common signaling methods used with FXS ports on Cisco Analog Gateways.

**Loop Start Signaling**

Loop Start Signaling is the most common technique for access signaling in a standard Public Switch Telephone Network (PSTN), or analog port connect a number of devices to your network. Most residential telephones are analog loop-start telephones, based upon the concept of the local loop you saw previously. The loop is an electrical communication path that consists of two wires, one to transmit and one to receive voice signals.

The two-wire circuit is still referred to as the **tip and ring**, with the tip tied to the ground and the ring tied to the negative side of the battery. When the phone handset is picked up (goes off hook), this action closes the circuit, and establishes a loop between the FXS port and the phone. Current is drawn from the battery of the analog port, which indicates a change in status. This change in status signals the current detector in the analog port to provide dial tone.

An incoming call is signaled to the handset by a standard on/off pattern, which causes the telephone to ring.

**VPM Signaling for Outbound Call**

To better understand what the logs look like for a successful outbound call on an FXS Port, these logs have been annotated so that you can clearly identify each portion of the call.

```plaintext
007578: Jul 2 09:15:50.655: %SYS-7-USERLOG_DEBUG: Message from tty867(user id: ): GOING OFF HOOK
007579: Jul 2 09:15:51.903: htsp_dsp_message: SEND_SIG_STATUS: state=0xC timestamp=62909 systime=6970515
007580: Jul 2 09:15:51.903: htsp_process_event: [0/3/0, FXSLS_ONHOOK, E_DSP_SIG_1100]fxsls_onhook_offhook htsp_setup_ind
007581: Jul 2 09:15:51.903: [0/3/0] get_local_station_id calling num= calling name= calling time=07/02 09:15 orig called=
007582: Jul 2 09:15:51.904: htsp_process_event: [0/3/0, FXSLS_WAIT_SETUP_ACK, E_HTPS_SETUP_ACK]fxsls_check_auto_call
007583: Jul 2 09:16:00.879: %SYS-7-USERLOG_DEBUG: Message from tty867(user id: ): DIALING 2002
007584: Jul 2 09:16:02.261: htsp_digit_ready(0/3/0): digit = 2
007585: Jul 2 09:16:02.734: htsp_digit_ready(0/3/0): digit = 0
```
007586: Jul 2 09:16:03.005: htsp_digit_ready(0/3/0): digit = 0
007587: Jul 2 09:16:03.438: htsp_digit_ready(0/3/0): digit = 2
007588: Jul 2 09:16:03.439: htsp_process_event: [0/3/0, FXSLS_OFFHOOK, E_HTSP_PROCEEDING]htsp_alert_notify

007589: Jul 2 09:16:08.241: %SYS-7-USERLOG_DEBUG: Message from tty867(user id: ): RING BACK
007590: Jul 2 09:16:10.621: htsp_call_bridged invoked
007591: Jul 2 09:16:10.665: htsp_process_event: [0/3/0, FXSLS_OFFHOOK, E_HTSP_CONNECT]fxsls_offhook_connect
007592: Jul 2 09:16:10.665: [0/3/0] nim_setSigState: ABCD=6, timestamp=0, sys_time=6972391
007593: Jul 2 09:16:10.665: [0/3/0] set signal state = 0x6 timestamp = 0
007594: Jul 2 09:16:10.667: htsp_process_event: [0/3/0, FXSLS_CONNECT, E_HTSP_VOICE_CUT_THROUGH]fxsls_voice_cut_thru

007595: Jul 2 09:16:20.815: %SYS-7-USERLOG_DEBUG: Message from tty867(user id: ): TWO WAY AUDIO

This document has also included what the signaling looks like for an inbound call as well. The logs have been annotated for ease of understand each step in the process.
Troubleshoot FXS and FXO Ports

Now that the basics have been covered for the different states of your analog ports, and what a call would look like in a perfect scenario, this document looks at the different ways to troubleshoot these ports. More specifically it looks at some show commands and some common failure scenarios.

Commands to Troubleshoot

Show Commands

To help troubleshoot what state the port is in, you can use commands like `show voice port summary` and `show voice call summary`. These commands show the different states such as when the call is on hook and not in use, to when the port is off hook and there is an active call. This figure, shows some of the different states.

On Hook:

Troubleshoot FXS and FXO Ports

Now that the basics have been covered for the different states of your analog ports, and what a call would look like in a perfect scenario, this document looks at the different ways to troubleshoot these ports. More specifically it looks at some show commands and some common failure scenarios.

Commands to Troubleshoot

Show Commands

To help troubleshoot what state the port is in, you can use commands like `show voice port summary` and `show voice call summary`. These commands show the different states such as when the call is on hook and not in use, to when the port is off hook and there is an active call. This figure, shows some of the different states.

On Hook:
008110: Jul 2 10:54:42.225: htsp_timer_stop3 htsp_setup_req
008111: Jul 2 10:54:42.225: Orig called num:88777
008112: Jul 2 10:54:42.225: htsp_process_event: [0/3/0, FXSLS_ONHOOK, E_HTSP_SETUP_REQ]fxsls_onhook_setuphtsp_alert
008113: Jul 2 10:54:42.225: [0/3/0] nim_set_sig_state: ABCD=0, timestamp=0, sys_time=7563547
008114: Jul 2 10:54:42.225: [0/3/0] set signal state = 0x0 timestamp = 0
008115: Jul 2 10:54:42.226: htsp_process_event: [0/3/0, FXSLS_WAIT_OFFHOOK, E_HTSP_VOICE_CUT_THROUGH]fxsls_waitoff_voice

008117: Jul 2 10:54:52.960: %SYS-7-USERLOG_DEBUG: Message from tty867(user id: ): PHONE GOES OFF HOOK
008118: Jul 2 10:54:55.431: htsp_dsp_message: SEND_SIG_STATUS: state=0xC timestamp=42727 systime=7564868
008119: Jul 2 10:54:55.431: htsp_process_event: [0/3/0, FXSLS_WAIT_OFFHOOK, E_DSP_SIG_1100]fxsls_offhook_dial htsp_dial
008120: Jul 2 10:54:55.431: [0/3/0] nim_set_sig_state: ABCD=4, timestamp=0, sys_time=7564868
008121: Jul 2 10:54:55.432: [0/3/0] set signal state = 0x4 timestamp = 0
008122: Jul 2 10:54:55.432: [0/3/0] nim_set_sig_state: ABCD=6, timestamp=200, sys_time=7564868
008123: Jul 2 10:54:55.432: [0/3/0] set signal state = 0x6 timestamp = 200
008124: Jul 2 10:54:55.432: htsp_timer2 - 200 msec
008125: Jul 2 10:54:55.631: htsp_process_event: [0/3/0, FXSLS_CONNECT, E_HTSP_EVENT_TIMER2]fxsls_conn_dial_done
008126: Jul 2 10:54:55.631: htsp_process_event: [0/3/0, FXSLS_CONNECT, E_DSP_DIALING_DONE]fxsls_conn_dial_done
008127: Jul 2 10:54:55.640: htsp_process_event: [0/3/0, FXSLS_CONNECT, E_HTSP_VOICE_CUT_THROUGH]fxsls_voice_cut_thru

008128: Jul 2 10:55:08.864: %SYS-7-USERLOG_DEBUG: Message from tty867(user id: ): TWO WAY AUDIO

008130: Jul 2 10:55:29.798: htsp_timer_stop3
008131: Jul 2 10:55:30.983: htsp_process_event: [0/3/0, FXSLS_RELEASE_REQ]fxsls_release_req
008132: Jul 2 10:55:30.983: htsp_process_event: [0/3/0, FXSLS_RELEASE_REQ]fxsls_release_req
008133: Jul 2 10:55:30.983: htsp_process_event: [0/3/0, FXSLS_RELEASE_REQ]fxsls_release_req
008134: Jul 2 10:55:30.983: [0/3/0] nim_set_sig_state: ABCD=12, timestamp=0, sys_time=7568309
008135: Jul 2 10:55:30.983: [0/3/0] set signal state = 0xC timestamp = 0
008136: Jul 2 10:55:30.983: [0/3/0] nim_set_sig_state: ABCD=4, timestamp=750, sys_time=7568309
008137: Jul 2 10:55:30.983: [0/3/0] set signal state = 0x4 timestamp = 750
008138: Jul 2 10:55:30.983: htsp_process_event: [0/3/0, FXSLS_CPC, E_HTSP_EVENT_TIMER]fxsls_cpc_timer
008139: Jul 2 10:55:30.983: htsp_process_event: [0/3/0, FXSLS_CPC, E_HTSP_EVENT_TIMER]fxsls_cpc_timer
008140: Jul 2 10:55:30.983: htsp_process_event: [0/3/0, FXSLS_CPC, E_HTSP_EVENT_TIMER]fxsls_cpc_timer
008141: Jul 2 10:55:30.983: htsp_process_event: [0/3/0, FXSLS_CPC, E_HTSP_EVENT_TIMER]fxsls_cpc_timer
008142: Jul 2 10:55:30.983: htsp_process_event: [0/3/0, FXSLS_CPC, E_HTSP_EVENT_TIMER]fxsls_cpc_timer
008143: Jul 2 10:55:30.983: htsp_process_event: [0/3/0, FXSLS_CPC, E_HTSP_EVENT_TIMER]fxsls_cpc_timer
008144: Jul 2 10:55:30.983: htsp_process_event: [0/3/0, FXSLS_CPC, E_HTSP_EVENT_TIMER]fxsls_cpc_timer


Phone Being Alerted:

008109: Jul 2 10:54:34.424: %SYS-7-USERLOG_DEBUG: Message from tty867(user id: ): PHONE IS IN IDLE & ON HOOK. THEN IT STARTS TELLING PHONE TO RING.
008111: Jul 2 10:54:52.960: %SYS-7-USERLOG_DEBUG: Message from tty867(user id: ): PHONE GOES OFF HOOK
008118: Jul 2 10:54:55.431: htsp_dsp_message: SEND_SIG_STATUS: state=0xC timestamp=42727 systime=7564868
008120: Jul 2 10:54:55.431: nim_set_sig_state: ABCD=4, timestamp=0, sys_time=7564868
008122: Jul 2 10:54:55.431: set signal state = 0x4 timestamp = 200
008124: Jul 2 10:54:55.431: htsp_timer2 - 200 msec
008126: Jul 2 10:54:55.631: htsp_process_event: [0/3/0, FXSLS_CONNECT, E_DSP_DIALING_DONE]fxsls_conn_dial_done
008137: Jul 2 10:55:29.843: set signal state = 0x4 timestamp = 750
008109: Jul 2 10:54:34.424: %SYS-7-USERLOG_DEBUG: Message from tty867(user id: ): PHONE IS IN IDLE & ON HOOK, THEN IT STARTS TELLING PHONE TO RING.
008115: Jul 2 10:54:42.226: htsp_call_bridged invoked
008116: Jul 2 10:54:42.227: htsp_process_event: [0/3/0, FXSLS_WAIT_OFFHOOK, E_HTSP_VOICE_CUT_THROUGH]fxsls_waitoff_voice

008117: Jul 2 10:54:52.960: %SYS-7-USERLOG_DEBUG: Message from tty867(user id: ): PHONE GOES OFF HOOK
008118: Jul 2 10:54:55.431: htsp_dsp_message: SEND_SIG_STATUS: state=0xC timestamp=42727 systime=7564868
008119: Jul 2 10:54:55.431: htsp_process_event: [0/3/0, FXSLS_WAIT_OFFHOOK, E_DSP_SIG_1100]fxsls_waitoff_offhook
008120: Jul 2 10:54:55.431: [0/3/0] nim_set_sig_state: ABCD=4, timestamp=0, sys_time=7564868
008121: Jul 2 10:54:55.431: [0/3/0] set signal state = 0x4 timestamp = 0
008122: Jul 2 10:54:55.431: [0/3/0] nim_set_sig_state: ABCD=6, timestamp=200, sys_time=7564868
008123: Jul 2 10:54:55.431: [0/3/0] set signal state = 0x6 timestamp = 200
008124: Jul 2 10:54:55.431: [0/3/0] htsp_timer2 - 200 msec
008125: Jul 2 10:54:55.431: htsp_process_event: [0/3/0, FXSLS_WAIT_OFFHOOK, E_HTSP_EVENT_TIMER2]fxsls_offhook_dial htsp_dial
008126: Jul 2 10:54:55.431: htsp_process_event: [0/3/0, FXSLS_CONNECT, E_DSP_DIALING DONE]fxsls_connc_disconnect
008127: Jul 2 10:54:55.431: htsp_process_event: [0/3/0, FXSLS_CONNECT, E_HTSP_VOICE_CUT_THROUGH]fxsls_voice_cut_thru

008128: Jul 2 10:55:08.864: %SYS-7-USERLOG_DEBUG: Message from tty867(user id: ): TWO WAY AUDIO
008129: Jul 2 10:55:27.232: %SYS-7-USERLOG_DEBUG: Message from tty867(user id): PHONE IS NOW DISCONNECTED FORM FAR END
008130: Jul 2 10:55:29.798: htsp_timer_stop3
008131: Jul 2 10:55:29.843: htsp_process_event: [0/3/0, FXSLS_CONNECT, E_HTSP_RELEASE_REQ]fxsls_connect_disc
008132: Jul 2 10:55:29.843: htsp_timer_stop
008133: Jul 2 10:55:29.843: [0/3/0] nim_set_sig_state: ABCD=12, timestamp=0, sys_time=7568309
008134: Jul 2 10:55:29.843: [0/3/0] set signal state = 0xC timestamp = 0
008136: Jul 2 10:55:29.843: [0/3/0] set signal state = 0x4 timestamp = 750
008137: Jul 2 10:55:29.843: htsp_timer - 950 msec fxsls_simulate_onhook
008138: Jul 2 10:55:30.793: htsp_process_event: [0/3/0, FXSLS_CPC, E_HTSP_EVENT_TIMER]fxsls_cpc_timer
008139: Jul 2 10:55:30.793: htsp_timer - 60000 msec
008140: Jul 2 10:55:30.808: htsp_dsp_message: RESP_SIG_STATUS: state=0xC timestamp=0 systime=7568405
008141: Jul 2 10:55:30.808: htsp_process_event: [0/3/0, FXSLS_WAIT_ONHOOK, E_DSP_SIG_1100]fxsls_waitonhook_offhook
008142: Jul 2 10:55:37.525: htsp_dsp_message: SEND_SIG_STATUS: state=0x4 timestamp=19285 systime=7569077
008143: Jul 2 10:55:37.525: htsp_process_event: [0/3/0, FXSLS_WAIT_ONHOOK, E_DSP_SIG_0100]fxsls_waitonhook_onhook

Among the previous two show commands, these could potentially be helpful to you in the future:

- show call active voice brief
- show voice call status
- show voice dsp active
- show voice dsp error
- show voice dsp group all

Voice Port Test Commands

Detector-Related Function Tests
The test voice port detector command, you are able to force a particular detector into an on or off state, perform tests on the detector, and then return the detector to its original state.

To configure this feature, enter these commands in privileged EXEC mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# test voice port slot/subunit/port detector {m-lead</td>
<td>battery-reversal</td>
</tr>
<tr>
<td>Router# test voice port slot/subunit/port detector {m-lead</td>
<td>battery-reversal</td>
</tr>
</tbody>
</table>

Loopback Function Tests

To establish loopbacks on a voice port, enter these commands in privileged EXEC mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# test voice port slot/subunit/port loopback {local</td>
<td>network}</td>
</tr>
<tr>
<td>Router# test voice port slot/subunit/port loopback disable</td>
<td>Identifies the voice port on which you want to end the test and enters the keyword disable to end the loopback.</td>
</tr>
</tbody>
</table>

Tone Injection Tests

To inject a test tone into a voice port, enter these commands in privileged EXEC mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# test voice port slot/subunit/port inject-tone {local</td>
<td>network} {1000hz</td>
</tr>
<tr>
<td>Router# test voice port slot/subunit/port inject-tone disable</td>
<td>Identifies the voice port on which you want to end the test and enter the keyword disable to end the test tone. Note: The disable keyword is only available if a test condition is already activated.</td>
</tr>
</tbody>
</table>
### Relay-Related Function Tests

To test relay-related functions on a voice port, enter these commands in privileged EXEC mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# test voice port slot/subunit/port relay {e-lead</td>
<td>loop</td>
</tr>
<tr>
<td>Router# test voice port slot/subunit/port relay {e-lead</td>
<td>loop</td>
</tr>
</tbody>
</table>

### Fax/Voice Mode Tests

The `test voice port switch fax` command forces a voice port into fax mode in order to test. After you enter this command, you can use the `show voice call` or `show voice call summary` command to check whether the voice port is able to operate in fax mode. If no fax data is detected by the voice port, the voice port remains in fax mode for 30 seconds and then reverts automatically to voice mode.

The `disable` keyword ends the forced mode switch; however, the fax mode ends automatically after 30 seconds. The `disable` keyword is available only while the voice port is in fax mode.

To force a voice port into fax mode and return it to voice mode, enter these commands in privileged EXEC mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# test voice port slot/subunit/port switch fax</td>
<td>Identifies the voice port you want to test. Enter the keyword fax to force the voice port into fax mode.</td>
</tr>
<tr>
<td>Router# test voice port slot/subunit/port switch disable</td>
<td>Identifies the voice port on which you want to end the test. Enter the keyword disable to return the voice port to voice mode.</td>
</tr>
</tbody>
</table>

### Common Issues Found

As mentioned at the start of this document, it covers some common issues found when you troubleshoot FXO and FXS.

**FXO Power Denial Detected**
The FXO is responsible to detect when power denial is done by the FXS, so that it knows when to go on-hook for FXS-side disconnect scenarios.

008109: Jul 2 10:54:34.424: %SYS-7-USERLOG_DEBUG: Message from tty867(user id: ): PHONE IS IN IDLE & ON HOOK. THEN IT STARTS TELLING PHONE TO RING.
008110: Jul 2 10:54:42.225: htsp_timer_stop3 htsp_setup_req
008111: Jul 2 10:54:42.225: Orig called num:88777
008112: Jul 2 10:54:42.225: [0/3/0, FXSLS_ONHOOK, E_HTSP_SETUP_REQ]fxsls_onhook_setuphtsp_alert
008113: Jul 2 10:54:42.225: [0/3/0] nim_set_sig_state: ABCD=0, timestamp=0, sys_time=7563547
008114: Jul 2 10:54:42.225: [0/3/0] set signal state = 0x0 timestamp = 0
008115: Jul 2 10:54:42.226: htsp_call_bridged invoked
008116: Jul 2 10:54:42.227: htsp_process_event: [0/3/0, FXSLS_WAIT_OFFHOOK, E_HTSP_VOICE_CUT_THROUGH]fxsls_waitoff_voice

008117: Jul 2 10:54:52.960: %SYS-7-USERLOG_DEBUG: Message from tty867(user id: ): PHONE GOES OFF HOOK
008118: Jul 2 10:54:55.431: htsp_dsp_message: SEND_SIG_STATUS: state=0xC timestamp=42727 systime=7564868
008119: Jul 2 10:54:55.431: [0/3/0, FXSLS_WAIT_OFFHOOK, E_DSP_SIG_1100]fxsls_waitoff_offhook
008120: Jul 2 10:54:55.431: [0/3/0] set signal state = 0x4 timestamp = 0
008121: Jul 2 10:54:55.431: [0/3/0] set signal state = 0x6 timestamp = 200
008122: Jul 2 10:54:55.432: [0/3/0] nim_set_sig_state: ABCD=12, timestamp=0, sys_time=7568309
008123: Jul 2 10:54:55.432: [0/3/0] set signal state = 0xC timestamp = 0
008124: Jul 2 10:54:55.432: [0/3/0] set signal state = 0x4 timestamp = 750
008125: Jul 2 10:54:55.631: htsp_process_event: [0/3/0, FXSLS_CONNECT, E_HTSP_VOICE_CUT_THROUGH]fxsls_voice_cut_thru

008126: Jul 2 10:55:08.864: %SYS-7-USERLOG_DEBUG: Message from tty867(user id: ): TWO WAY AUDIO


008128: Jul 2 10:55:08.864: %SYS-7-USERLOG_DEBUG: Message from tty867(user id: ): TWO WAY AUDIO


008130: Jul 2 10:55:29.798: htsp_timer_stop3
008131: Jul 2 10:55:29.843: [0/3/0, FXSLS_CONNECT, E_HTSP_RELEASE_REQ]fxsls_connect_disc
008132: Jul 2 10:55:29.843: htsp_timer_stop
008133: Jul 2 10:55:29.843: [0/3/0] nim_set_sig_state: ABCD=12, timestamp=0, sys_time=7568309
008134: Jul 2 10:55:29.843: [0/3/0] set signal state = 0xC timestamp = 0
008136: Jul 2 10:55:29.843: [0/3/0] set signal state = 0x4 timestamp = 750
008137: Jul 2 10:55:29.843: htsp_timer - 950 msec
008138: Jul 2 10:55:30.793: [0/3/0, FXSLS_CPC, E_HTSP_EVENT_TIMER]fxsls_cpc_timer
008139: Jul 2 10:55:30.793: htsp_timer - 60000 msec
008140: Jul 2 10:55:30.808: [0/3/0, FXSLS_WAIT_ONHOOK, E_DSP_SIG_1100]fxsls_wait_onhook
008141: Jul 2 10:55:30.808: htsp_dsp_message: SEND_SIG_STATUS: state=0x4 timestamp=19285 systime=7569077
008142: Jul 2 10:55:30.808: htsp_dsp_message: SEND_SIG_STATUS: state=0xC timestamp=0
008143: Jul 2 10:55:37.525: [0/3/0, FXSLS_WAIT_OFFHOOK, E_DSP_SIG_0100]fxsls_waitoff_onhook
008144: Jul 2 10:55:37.525: [0/3/0] nim_set_sig_state: ABCD=4, timestamp=750, sys_time=7569077
008145: Jul 2 10:55:37.525: [0/3/0] set signal state = 0xC timestamp = 0

The fxols_power_denial_detected event is triggered when there is no loop current detected on
By default, there is a 750msec timer started. If the DSP does not detect current before the timer expires, it disconnects the call. The timer can be modified under the voice-port config mode with the `timeouts power-denial <0-2500ms>` command. This timer must match what the FXS side has defined for their power denial duration.

This scenario indicates either faulty a cable, hardware, or wrong port type on the other side. Determine if the problem follows the port or the line.

- If the problem follows the line, have them check the cables up to the telco demark. Engage the telco for assistance.
- If the problem follows the port, then it's likely a faulty port. This can be further troubleshoot to confirm.

**Other Gotchas**

- **Disconnect supervision** - Review the specific section for how disconnect supervision is handled on analog ports.
- **Wiring** - Wiring must be 2-wire, straight-through, from FXS to FXO. If you fail to get dial tone with a wiring issue, you usually hear absolutely no audio on the line. If the cable connected properly, you can hear a slight increase in the noise floor when you go off-hook.
- **Bad port** - Ports can go bad, and fail to give dial-tone, detect ringing voltage, etc. Troubleshoot to isolate the port from the VoIP side and from the cable side.
- **DSP issues** - The port needs to use a DSP to identify events on the port. Hence, voice-ports allocate DSPs for signaling at boot, even if the port is not in use and is shut-down. When you make changes to analog voice-ports, shut/no shut the port before you test again.
- **Long/poor runs, impedance issues** - Since it is analog audio transmitted, the health of Electromagnetic interference (EMI) in the environment is important since it can impact audio quality. For example, when you run your analog lines over a fluorescent light (or near a blender/motor/etc), this can cause excessive noise on the line. Long runs in general cause attenuation and impedance mis-match. Proper impedance must be set for the run length.
- **Excessive gain to compensate for attenuation** - When you apply high amounts of input gain, this can exacerbate echo issues since it causes low echo return loss (ERL). Try to avoid this when possible.
- **Digit delivery** - Digits are not sent from the telco to an FXO port. You must use `connection plar <extension>` to route the call from the port to either a receptionist, or an IVR/AA.
- **Outgoing call failure**? If the circuit is FXOGS and you have it configured for FXOLS, incoming calls work, but outbound fails. Also, polarity is important for outgoing calls with GS.