

Using IVR to Give Basic DID Capability for AVVID

Document ID: 7202

Introduction

Prerequisites

- Requirements
- Components Used
- Conventions

Configure

- Network Diagram
- Configurations

Verify

Troubleshoot

- Troubleshoot Commands

Related Information

Introduction

If the Telco is able to supply them, direct inward dial (DID) exchange trunks allow a caller to ring certain fixed leading digits and a variable extension range. This allows the exchange to pass certain digits to the attached private automatic branch exchange (PABX). The PABX matches the digits to internal extensions. For example, a customer is given the digit range of 8448 7000 through 8448 8000. This gives a total extension range of 7000 to 8000. When any number in this range is called, the exchange seizes a trunk. It passes the last four digits as dual-tone multifrequency relay (DTMF) tones to the connected PABX. DID trunks need special hardware at the exchange and on the PABX, as they have to recognize the incoming call from the exchange and then accept the DTMF tones the exchange passes towards them before they attempt to ring the called extension.

Normal Cisco Foreign Exchange Office (FXO) ports are unable to accept the DTMF digits upon initial seizure. Therefore, they are not able to support the DID function. This document explains a workaround to allow a basic DID facility through FXO ports. This is accomplished by using IP-powered interactive voice response (IP IVR) functionality to play an audio prompt that requests the caller to enter the required extension or operator number.

Note: Cisco IP Auto Attendant performs the same function on a Cisco CallManager server. Refer to *Configuring the Cisco Four Port IP AutoAttendant Application on a Cisco CallManager Server* for further information.

Prerequisites

Requirements

Before you attempt this configuration, ensure that you meet these prerequisites:

- A basic understanding of VoIP, Cisco CallManager, and IVR.

Components Used

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

- Cisco IOS® Software Release 12.3(4)T1
- Cisco 3725 Router
- NM-2V
- Voice interface card (VIC)-2FXO

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, make sure that you understand the potential impact of any command.

Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

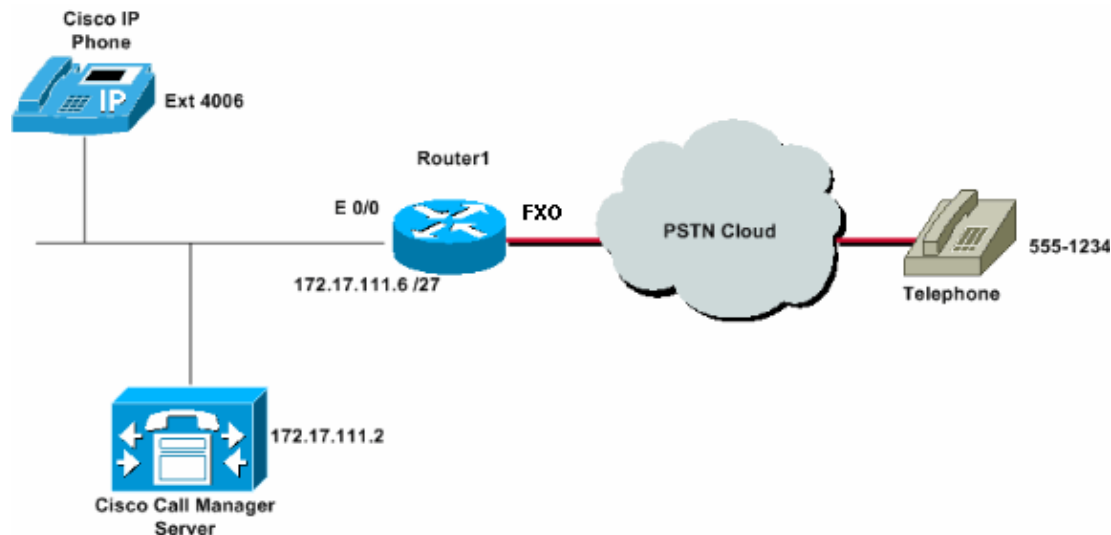
Configure

In this section, you are presented with the information to configure the features described in this document.

Note: To find additional information on the commands used in this document, use the Command Lookup Tool (registered customers only).

Network Diagram

This document uses the network setup shown in this diagram:



Configurations

This is a sample configuration that shows how to set up a router as a gateway between the public switched telephone network (PSTN) and a Cisco CallManager server. The Cisco CallManager server extensions are in the range 4000 – 4999 and an attendant number of 0 configured.

The audio prompt is stored on the router Flash memory with a name of **enter_destination.au**. Audio prompts are generic 8 khz, 8-bit u-law coding Sun audio format (stored as file name *.AU). Therefore, messages can be customized and saved to the router's Flash through TFTP. The text of the test message is:

If you know the extension, please dial it now, or dial 0 for the attendant

The IVR application used is the **clid_authen_col_npw** tcl script. This is built into Cisco IOS. Use the command **show call application voice clid_authen_col_npw** to view the complete script in the router. To obtain more information regarding various IVR applications, refer to Configuring Interactive Voice Response for Cisco Access Platforms. The caller is authenticated locally with a null password and a dummy destination pattern which corresponds to a local username. This configuration segment shows how this is accomplished:

```
!  
username 999  
authentication login h323 local  
!  
dial-peer voice 999 pots  
  application clid_authen_col_npw  
  destination-pattern 999  
  port 2/1/0  
!
```

This is a complete working configuration.

Router 1
Router1# Current configuration: ! version 12.3 service timestamps debug uptime service timestamps log uptime no service password-encryption ! hostname Router1 ! ! username 999 authentication login h323 local <i>!--- Username to authenticate locally.</i> ! ip subnet-zero ! ! interface Ethernet0/0 ip address 172.17.111.6 255.255.255.224 ! interface Ethernet0/1 no ip address shutdown ! ip classless ip route 0.0.0.0 0.0.0.0 172.17.111.1 no ip http server ! ! voice-port 2/0/0 ! voice-port 2/0/1 ! voice-port 2/1/0 cptone AU timing percentbreak 60 ! voice-port 2/1/1 cptone AU

```
    timing percentbreak 60
!
dial-peer voice 998 pots
  application clid_authen_col_npw

!--- Use the IVR tcl script.

  destination-pattern 999

!--- Destination pattern matches username.

  port 2/1/0

!--- Bind to FXO port 2/1/0.

!
dial-peer voice 999 pots
  application clid_authen_col_npw
  destination-pattern 999
  port 2/1/1
!
dial-peer voice 2000 pots

!--- Peers 2000 and 2001 are for outdial.

  destination-pattern 9
  port 2/1/0
!
dial-peer voice 2001 pots
  destination-pattern 9
  port 2/1/1
!
dial-peer voice 4000 voip

!--- Allows calls to CallManager extensions.

  destination-pattern 4...
  session target ipv4:172.17.111.2

!--- IP address of CallManager.

  codec g711ulaw

!--- Use the correct codec.

!
dial-peer voice 1001 voip

!--- Allows calls to CallManager attendant.

  destination-pattern 0
  session target ipv4:172.17.111.2
  codec g711ulaw
!
!
line con 0
  transport input none
line aux 0
line vty 0 4
  login
!
end
```

Verify

This section provides information to confirm that your configuration works properly.

Certain **show** commands are supported by the Output Interpreter Tool (registered customers only) . This allows you to view an analysis of **show** command output.

- **show flash** Displays the contents of Flash memory and shows if the audio file enter_destination.au is in the Flash.
- **show call active voice brief** Displays the call status for all voice ports.
- **show debugging** Displays which **debug** commands are enabled at any given time.
- **show call application voice summary** – Displays a list of the voice applications that are configured.

After you enter the configurations shown above into your router, verify that the network operates correctly. These commands and their respective output show you a successful implementation of the configurations above.

The most common problems with IVR include:

- The application is not properly configured on the plain old telephone service (POTS) dial-peer.
- Make sure that the incoming telephony call matches the POTS dial-peer with the application. The POTS dial-peer matches the first dial-peer with the matching **port** command in the configuration or if the called number matches the dial-peer subcommand **incoming called-number <number>** .
- Check that the application actually exists through the **show call application voice summary** command. Make sure that the application name does not contain a syntax error.
- Make sure that DID is not configured on the POTS dial-peer.
- Ensure that the proper audio files have been loaded into Flash.
- Ensure that the authentication with username and password has been properly configured.

```
Router1#show flash:
System flash directory:
  File Length Name/status
  1 11316708 c3640-is-mz.122-19.bin
  2 43548 enter_destination.au
[11360384 bytes used, 21669760 available, 33030144 total]
32768K bytes of processor board System flash (Read/Write)
```

```
Router1#show call active voice
Total call-legs: 2
GENERIC:
  SetupTime=5643862 ms
  Index=1
  PeerAddress=555-1234
```

!--- Output is suppressed.

```
ConnectTime=5643863
CallDuration=00:00:55
CallState=4
CallOrigin=2
ChargedUnits=0
InfoType=2
TransmitPackets=1988
TransmitBytes=318080
ReceivePackets=1868
ReceiveBytes=298880
TELE:
ConnectionId=[0xEFDD67C7A 0x157111CC 0x802CAFBE 0xB1BDDE4D]
IncomingConnectionId=[0xEFDD67C7A 0x157111CC 0x802CAFBE 0xB1BDDE4D]
```

```
!--Output suppressed
GENERIC:
SetupTime=5644387 ms
Index=1
PeerAddress=4006
```

!--- Output is suppressed.

```
ConnectTime=5645637
CallDuration=00:00:40
CallState=4
CallOrigin=1
ChargedUnits=0
InfoType=2
TransmitPackets=2003
TransmitBytes=320480
ReceivePackets=2002
ReceiveBytes=320320
VOIP:
ConnectionId[0xEF67C7A 0x157111CC 0x802CAFBE 0xB1BDDE4D]
IncomingConnectionId[0xEF67C7A 0x157111CC 0x802CAFBE 0xB1BDDE4D]
RemoteIPAddress=172.17.111.2
RemoteUDPPort=31200
RemoteSignallingIPAddress=172.17.111.2
RemoteSignallingPort=1720
```

!--- Output is suppressed.

```
SessionProtocol=cisco
SessionTarget=ipv4:172.17.111.2
OnTimeRvPayout=35825
```

!--- Output is suppressed.

```
VAD = enabled
CoderTypeRate=g711ulaw
CodecBytes=160
```

Total call-legs: 2

```
Router1#show call active voice brief
<ID>: <start>hs.<index> +<connect> pid:<peer_id> <dir> <addr> <state>
dur hh:mm:ss tx:<packets>/<bytes> rx:<packets>/<bytes>
IP <ip>:<udp> rtt:<time>ms pl:<play>/<gap>ms lost:<lost>/<early>/<late>
delay:<last>/<min>/<max>ms <codec>
```

!--- Output is suppressed.

Total call-legs: 2

```
11F8 : 5643862hs.1 +1 pid:2 Answer 555-1234 active
dur 00:01:04 tx:2440/390400 rx:2320/371200
Tele port 2/1/0:24: tx:46400/46400/0ms g711ulaw noise:0 acom:14 i/0:-26/-42 dBm

11F8 : 5644387hs.1 +1250 pid:3 Originate 4006 active
dur 00:00:47 tx:2377/380320 rx:2376/380160
IP 10.77.241.159:31200 rtt:0ms pl:44995/0ms lost:0/0/0 delay:54/54/65ms g711ulaw
```

Total call-legs: 2

```
Router1#show debugging
voip:
voip ccAPI function enter/exit debugging is on
ivr:
```

```

ivr errors debugging is on
ivr state transitions debugging is on
ivr settlement activities debugging is on
ivr script debugging is on
ivr app library debugging is on
ivr tcl commands debugging is on
ivr digit collect debugging is on
ivr call setup debugging is on
Router1#

```

Troubleshoot

This section provides the information to troubleshoot your configuration.

Troubleshoot Commands

Certain **show** commands are supported by the Output Interpreter Tool (registered customers only). This allows you to view an analysis of **show** command output.

These are the **debug** commands which can be used in this scenario.

Note: Before you issue **debug** commands, refer to the Important Information on Debug Commands.

- **debug voip ccapi inout** Used to debug the call control application program interface (API).
- **debug voip ivr** Used to debug the IVR application.

```
Router1#debug voip ccapi inout
```

```

*Mar 1 16:15:48.659: cc_api_call_setup_ind (vdbPtr=0x621AC08C, callInfo={called
=, called_oct3=0x81, calling=, calling_oct3=0x0, calling_oct3a=0x0,
calling_xlated=false, subscriber_type_str=RegularLine, fdest=0, peer_tag=2,
prog_ind=3, callingIE_present 0}, callID=0x62004D2C)
*Mar 1 16:15:48.663: cc_api_call_setup_ind calling number is null,
answer addr dest pattern 999 e164_ans_addr 0 e164_dest_pattern 1
*Mar 1 16:15:48.663: cc_api_call_setup_ind valid dest pattern,
copying 999 to calling number
*Mar 1 16:15:48.663: cc_api_call_setup_ind type 2 , prot 0
*Mar 1 16:15:48.663: cc_process_call_setup_ind (event=0x6217B7E0)
*Mar 1 16:15:48.663: >>>>CCAPI handed cid 26 with tag 2 to app
"clid_authen_col_npw"
*Mar 1 16:15:48.663: ccCallAppHandoff (callID=0x1A)
*Mar 1 16:15:48.667: ccCallSetContext (callID=0x1A, context=0x6200E988)
*Mar 1 16:15:48.667: ccCallSetupAck (callID=0x1A)
*Mar 1 16:15:48.667: ccCallProceeding (callID=0x1A, prog_ind=0x0)
*Mar 1 16:15:48.667: ccCallConnect (callID=0x1A), prog_ind =
0ccCallConnect:setting callEntry->connected to TRUE

```

!--- Output is suppressed.

```

*Mar 1 16:16:00.675: cc_api_voice_mode_event , callID=0x1A
*Mar 1 16:16:00.675: Call Pointer =6216BAF0
*Mar 1 16:16:00.675: ccCallAppReturn (callID=0x1A)
*Mar 1 16:16:00.675: cc_process_notify_bridge_done (event=0x621815FC)
*Mar 1 16:16:00.675: ccCallSetContext (callID=0x1B, context=0x6200E988)
*Mar 1 16:16:02.711: cc_api_call_disconnected(vdbPtr=0x62072604, callID=0x1B,
cause=0x10)
*Mar 1 16:16:02.715: ccConferenceDestroy (confID=0x7, tag=0x0)
*Mar 1 16:16:02.719: ccCallDisconnect: existing_cause = 0x0, new_cause = 0x10
*Mar 1 16:16:02.723: cc_api_call_disconnect_done(vdbPtr=0x621AC08C,
callID=0x1A, disp=0, tag=0x0)

```

```
Router1#
Router1#
Router1#
Router1#
```

```
Router1#debug voip ivr
```

```
Basic ivr debugging is on
```

```
Router1#
```

```
*Mar 1 16:18:15.93
*Mar 1 16:18:15.931: App clid_authen_col_npw: Handling callID 28
*Mar 1 16:18:15.931: callingNumber=999, calledNumber=,
  redirectNumber= display_info= calling_oct3a=0
*Mar 1 16:18:15.931: accountNumber=, finalDestFlag=0,
  guid=314c.5398.1577.11cc.8034.afbe.blbd.de4d
*Mar 1 16:18:15.931: peer_tag=2
*Mar 1 16:18:15.931: tcl_app:no ssInfo in SETUP
*Mar 1 16:18:15.931: No language defined for the script yet
*Mar 1 16:18:15.931: settlement_validate_call: cid(28), target=, tokenp=0x0
  *Mar 1/16:18:15.931: : *Mar 1 16:18:15.935:
  Accepting CallID=28 *Mar 1 16:18:15.935: :/ani/
*Mar 1 16:18:15.935: :[authenticate]
*Mar 1 16:18:15.935: authenticate
*Mar 1 16:18:15.935: account=999
*Mar 1 16:18:15.939: ta_get_event: out of while loop for event authenticated
*Mar 1 16:18:15.939: cid(28) ta_get_event returning authenticated
*Mar 1 16:18:15.939: :[callID]
```

```
!--- Output is suppressed.
```

```
*Mar 1 16:18:15.943: cid(28) app running state get_dest
*Mar 1 16:18:15.947: ta_PromptCmd. CallID=28
*Mar 1 16:18:15.951: prompt and collect app got callID 28
*Mar 1 16:18:15.951: Playing prompt enter_destination.au
*Mar 1 16:18:15.951: Prompt interrupt enabled
*Mar 1 16:18:15.951: No return on play complete
*Mar 1 16:18:15.951: Matching against dial plan
*Mar 1 16:18:15.951: Abort key is * Termination key is #
*Mar 1 16:18:15.951: Matching against 0 patterns.
*Mar 1 16:18:15.951: ms_create() Iniz ply_timer
*Mar 1 16:18:15.951: act_reportDigitsDone callid=28 Enable succeeded.
*Mar 1 16:18:15.951: ms_associateDone: callID=0x1C, pVdb=0x621AC08C,
  disposition=0, playFunc=0x60EC9C54,
```

```
!--- Output is suppressed.
```

```
*Mar 1 16:18:15.955: ms_associateDone: 16:18:15.960, Tstart(ply: iSndDly 1000)
*Mar 1 16:18:16.955: $ $ms_process() >>ms_start_play()
*Mar 1 16:18:16.955: ms_start_play() stream->timeTry=0
*Mar 1 16:18:16.955: ms_start_play() 16:18:16.960 mgdTstop(ply)
*Mar 1 16:18:19.379: ms_stop_play() call 28 mgdTstop at 16:18:19.384
(cause MS STOP_COMPLETE)
*Mar 1 16:18:19.379: ms_stop_play(): Play Stopped at 16:18:19.384
*Mar 1 16:18:19.379: ms_stop_play: callID 28: calling CBF for stream_id 0
*Mar 1 16:18:19.379: $ $pcapp_playComplete():
*Mar 1 16:18:19.931: pcapp CallID 28 event CC_EV_CALL_DIGIT_BEGIN ignored digit=4
*Mar 1 16:18:20.491: pcapp CallID 28 event CC_EV_CALL_DIGIT_BEGIN ignored digit =0
*Mar 1 16:18:20.911: pcapp CallID 28 event CC_EV_CALL_DIGIT_BEGIN ignored digit =0
*Mar 1 16:18:21.671: pcapp CallID 28 event CC_EV_CALL_DIGIT_BEGIN ignored digit =6
*Mar 1 16:18:21.811: pcapp CallID 28 returning PCAPP_MATCHED. string=4006
*Mar 1 16:18:21.811: ta_get_event: out of while loop for event collect success
*Mar 1 16:18:21.811: cid(28) ta_get_event returning collect success
*Mar 1 16:18:21.815: :[callID]
*Mar 1 16:18:21.815: :/puts/
```

```
*Mar 1 16:18:21.815: cid(28) app running state place_call
*Mar 1 16:18:21.819: :[placeCall]
*Mar 1 16:18:21.819: Placing call for callID 28 to destination=4006
*Mar 1 16:18:21.819: placecall CallID 28 got event CC_EV_CALL_HANDOFF
*Mar 1 16:18:21.819: Matched peers(1)
*Mar 1 16:18:21.819: placecall pc_setupPeer cid(28), destPat(4006), matched(1),
prefix(), peer(620084A8)
*Mar 1 16:18:21.819: pcSettlementMatrixCheck: retcode=1 cid(28) trans=0x0,
provider=0 No settle-call present
*Mar 1 16:18:21.819: placecall cid(28) state change PC_CS_INIT to
PC_CS_CALL_SETTING
*Mar 1 16:18:21.907: placecall CallID 29 got event CC_EV_CALL_PROCEEDING
*Mar 1 16:18:21.919: placecall CallID 29 got event CC_EV_CALL_ALERT
*Mar 1 16:18:21.919: placecall cid(28) state change PC_CS_CALL_SETTING
to PC_CS_ALERT_RCVD
*Mar 1 16:18:29.495: placecall CallID 29 got event CC_EV_CALL_CONNECTED
*Mar 1 16:18:29.495: placecall cid(29) state change PC_CS_ALERT_RCVD
to PC_CS_CONFERENCING
*Mar 1 16:18:29.495: placecall CallID 29 got event CC_EV_CONF_CREATE_DONE
*Mar 1 16:18:29.495: placecall CallID 28 returning PLACECALL_ACTIVE.
*Mar 1 16:18:29.495: pCall(0x6200E988), settlement_credit_time=0
```

!--- Output is suppressed.

```
*Mar 1 16:18:37.211: cid(0) ta_get_event returning incoming disconnected
*Mar 1 16:18:37.211: ta_WaitEventCmd(): ta_get_event(event
[incoming disconnected])
*Mar 1 16:18:37.215: TCL script eval for callID 28 completed. code=OK
*Mar 1 16:18:37.215: incoming disconnected
Router1#
```

Related Information

- [IVR Data Sheet](#)
- [Cisco IP IVR Installation Guide](#)
- [Configuring Interactive Voice Response for Cisco Access Platforms](#)
- [VoIP with IVR](#)
- [Cisco IOS Voice, Video, and Fax Command Reference](#)
- [Deploying Cisco IP IVR Applications with the Cisco Application Editor](#)
- [Marketing IVR Homepage](#)
- [Integrating the Cisco IP IVR in the Cisco IPCC System](#)
- [Voice Technology Support](#)
- [Voice and Unified Communications Product Support](#)
- [Recommended Reading: Troubleshooting Cisco IP Telephony](#)
- [Technical Support & Documentation – Cisco Systems](#)

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)