



Configuring Interactive Voice Response

This chapter shows you how to configure Interactive Voice Response (IVR). This chapter contains the following sections:

- IVR Overview
- IVR Prerequisite Tasks
- Configuring IVR
- IVR Configuration Example

For a complete description of the commands used in this chapter, refer to the *Cisco IOS Multiservice Applications Command Reference* publication. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

IVR Overview

Interactive Voice Response (IVR) consists of simple voice prompting and digit collection to gather caller information for authenticating the user and identifying the destination. IVR applications can be assigned to specific ports or invoked based on DNIS. An IP PSTN gateway can have several different IVR applications to accommodate many different gateway services, and you can customize the IVR applications to present different interfaces to the different callers.

IVR uses Tool Control Language (TCL) scripts to gather information and to process accounting and billing. For example, a TCL IVR script plays when a caller receives a voice-prompt instruction to enter a specific type of information, such as a PIN. After playing the voice prompt, the IVR application collects the predetermined number of touch tones (digit collection) and forwards the collected digits to a server for storage and retrieval. Call records can be kept and a variety of accounting functions performed.



Note

All IVR scripts are modified and secured with a proprietary Cisco locking mechanism. Only Cisco internal technical support personnel can open and modify these scripts.

Cisco provides the following IVR scripts:

- `fax_hop_on_1`—Collects digits from the redialer, such as account number and destination number. When a call is placed to an H.323 network, the set of fields configured in the call information structure are *entered*, *destination*, and *account*.
- `clid_authen`—Authenticates the call with ANI and DNIS numbers, collects the destination data, and makes the call.

- `clid_authen_npw`—Same as `clid_authen`, but uses a null password when authenticating, rather than DNIS numbers.
- `clid_authen_collect`—Authenticates the call with ANI and DNIS numbers and collects the destination data, but if authentication fails, it collects the account and password.
- `clid_authen_col_npw`—Same as `clid_authen_collect`, but uses a null password and does not use or collect DNIS numbers.
- `clid_col_npw_3`—Same as `clid_authen_col_npw` except with that script, if authentication with the digits collected (account and PIN) fails, the `clid_authen_col_npw` script just plays a failure message (`auth_failed.au`) and then hangs up. The `clid_col_npw_3` script allows two failures, then plays the retry audio file (`auth_retry.au`) and collects the account and PIN again.

The caller can interrupt the message by entering digits for the account number, which triggers the prompt to tell the caller to enter the PIN. If authentication fails the third time, the script plays the audio file `auth_fail_final.au`, and hangs up.

Table 11 lists the prompt audio files associated with the `clid_col_npw_3` script.

Table 11 *clid_col_npw_3 Script Prompt Audio Files*

Prompt Audio Filename	Action
<code>flash:enter_account.au</code>	Asks the caller to enter an account number the first time.
<code>flash:auth_fail_retry.au</code>	Played after two failures, asks the caller to reenter the account number.
<code>flash:enter_pin.au</code>	Asks the caller to enter a PIN.
<code>flash:enter_destination.au</code>	Asks the caller to enter a destination phone number.
<code>flash:auth_fail_final.au</code>	Informs the caller that the authorization failed three times.

Table 12 lists additional audio files associated with the `clid_col_npw_3` script.

Table 12 *Additional clid_col_npw_3 Script Audio Files*

Script Audio Filename	Action
<code>auth_fail_retry.au</code>	Informs the caller that authorization failed. Prompts the caller to reenter the account number followed by the pound sign (#).
<code>auth_fail_final.au</code>	Informs the caller, “I’m sorry, your account number cannot be verified. Please hang up and try again.”

- `clid_col_npw_npw`—Tries to authenticate by using ANI, null as the user ID, user, and user password pair. If that fails, it collects an account number and authenticates with account and null. It allows three tries for the caller to enter the account number before ending the call with the authentication failed audio file. If authentication succeeds, it plays a prompt to enter the destination number.

Table 13 lists the audio files associated with the `clid_col_npw_npw` script.

Table 13 *clid_col_npw_npw Script Audio Files*

Script Audio Filename	Action
flash:enter_account.au	Asks the caller to enter the account number the first time.
flash:auth_fail_retry.au	Played after first two failures, asks the caller to reenter the account number.
flash:enter_destination.au	Asks the caller to enter the destination phone number.
flash:auth_fail_final.au	Informs the caller that the authorization failed three times.

- `clid_col_dnis_3.tcl`—Authenticates the caller ID three times. First it authenticates the caller ID with DNIS. If that is not successful, it attempts to authenticate with the caller PIN up to three times.
- `clid_col_npw_3.tcl`—Authenticates with null. If authentication is not successful, it attempts to authenticate by using the caller PIN up to 3 times.
- `clid_4digits_npw_3.tcl`—Authenticates with null. If the authentication is not successful, it attempts to authenticate with the caller PIN up to 3 times using the 14-digit account number and password entered together.
- `clid_4digits_npw_3_cli.tcl`—Authenticates the account number and PIN respectively by using ANI and null. The length of digits allowed for the account number and password are configurable through the CLI. If the authentication fails, it allows the caller to retry. The retry number is also configured through the CLI.
- `clid_authen_col_npw_cli.tcl`—Authenticates the account number and PIN respectively using ANI and null. If the authentication fails, it allows the caller to retry. The retry number is configured through the CLI. The account number and PIN are collected separately.
- `clid_authen_collect_cli.tcl`—Authenticates the account number and PIN by using ANI and DNIS. If the authentication fails, it allows the caller to retry. The retry number is configured through the CLI. The account number and PIN are collected separately.
- `clid_col_npw_3_cli.tcl`—Authenticates by using ANI and null for account and PIN respectively. If the authentication fails, it allows the caller to retry. The retry number is configured through the CLI.
- `clid_col_npw_npw_cli.tcl`—Authenticates by using ANI and null for account and PIN respectively. If authentication fails, it allows the caller to retry. The retry number is configured through the CLI. The account number and PIN are collected together.

**Note**

To see the contents of the TCL IVR script, use the **show call application voice** command.

IVR Prerequisite Tasks

Before you configure your Cisco gateway to support IVR, you need to perform the following prerequisite tasks:

- Configure VoIP to support H.323-compliant gateways—meaning that in addition to the basic configuration tasks, such as configuring dial peers and voice ports, you must configure specific devices in your network to act as gateways.
- Configure a TFTP sever to perform storage and retrieval of the audio files, which are required by the Debit Card gateway or other features requiring TCL IVR scripts and audio files.

- Download the appropriate classic or TCL IVR script from the CCO Software Support Center. Use the **copy** command to copy your audio file (.au file) to your Flash memory, and the **audio-prompt load** command to read it into RAM. For more information about loading files into Flash memory, see the “Managing Cisco AS5300 VFCs” section earlier in this chapter.
- Make sure that your audio files are in the proper format. The IVR prompts require audio file (.au) format of 8-bit, u-law, and 8-Khz encoding. To encode your own audio files, we recommend that you use one of these two audio tools (or a similar tool of comparable quality):
 - Cool Edit, manufactured by Syntrillium Software Corporation
 - AudioTool, manufactured by Sun Microsystems
- Make sure that your access platform has a minimum of 16 MB Flash and 64 MB of DRAM memory.
- Install and configure the appropriate RADIUS security server in your network. The version of RADIUS that you are using must be able to support IETF-Supported VSAs, which are implemented by using IETF RADIUS Attribute 26.

Configuring IVR

To configure IVR functionality using either classic or TCL scripts, perform the following tasks after you have completed the prerequisite steps:

- Create an application that will interact with the appropriate classic or TCL script.
- Define and pass the defined parameter values to the application. Depending on the TCL script you select, these values can include the language of the audio file and the location of the audio file. Table 14 lists the TCL scripts and the parameter values they require.
- Associate the application to the incoming POTS dial peer.
- Define the appropriate method lists using AAA so that you identify RADIUS as the security protocol performing accounting.

To configure IVR, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# call application voice <i>application-name location</i>	Defines the name to be used for your application and indicates the location (URL) of the appropriate IVR script to be used with this application.
Step 3	Router(config)# call application voice <i>application-name language language</i>	(Optional depending on the TCL script you select) Defines the language of the audio file for the designated application and passes that information to the application.
Step 4	Router(config)# call application voice <i>application-name pin-length number</i>	(Optional depending on the TCL script you select) Defines the number of characters in the PIN for the designated application and passes that information to the application.
Step 5	Router(config)# call application voice <i>application-name retry-count number</i>	(Optional depending on the TCL script you select) Defines the number of times a caller is permitted to reenter the PIN for the designated application and passes that information to the application.


	Command	Purpose
Step 6	Router(config)# call application voice <i>application-name uid-length number</i>	(Optional depending on the TCL script you select) Defines the number of characters in the UID for the designated application and passes that information to the application.
Step 7	Router(config)# call application voice <i>application-name set-location language category location</i>	(Optional depending on the TCL script you select) Defines the location, language, and category of the audio files for the designated application and passes that information to the application.
Step 8	Router(config)# aaa new-model	Enables AAA security and accounting services.
Step 9	Router(config)# gw-accounting h323	Enables gateway-specific H.323 accounting.
Step 10	Router(config)# aaa authentication login h323 radius	Defines a method list called h323 where RADIUS is defined as the only method of login authentication.
Step 11	Router(config)# aaa accounting connection h323 start-stop radius	Defines a method list called h323 where RADIUS is used to perform connection accounting, providing start-stop records.
Step 12	Router(config)# radius-server host ip-address <i>auth-port number acct-port number</i>	Identifies the RADIUS server and the ports that will be used for authentication and accounting services.
Step 13	Router(config)# radius-server key key	Specifies the password used between the gateway and the RADIUS server.
Step 14	Router(config)# dial-peer voice number pots	Enters the dial-peer configuration mode to configure the incoming POTS dial peer.  Note The <i>number</i> value of the dial-peer voice pots command is a tag that uniquely identifies the dial peer.
Step 15	Router(config-dial-peer)# application <i>application-name</i>	Associates the IVR application with the incoming POTS dial peer.
Step 16	Router(config-dial-peer)# destination-pattern <i>[+]string t</i>	Defines the telephone number associated with this dial peer.
Step 17	Router(config-dial-peer)# port port number	Defines the voice port associated with this dial peer.

Table 14 lists TCL script names and the corresponding parameters that are required for each TCL scripts.

Table 14 TCL Scripts and Parameters

TCL Script Name	Description—Summary	Commands to Configure
clid_4digits_npw_3_cli.tcl	This script authenticates the account number and PIN respectively using ANI and null. The length of digits allowed for the account number and password are configurable through the CLI. If the authentication fails, it allows the caller to retry. The retry number is also configured through the CLI.	call application voice uid-len min = 1, max = 20, default = 10 call application voice pin-len min = 0, max = 10, default = 4 call application voice retry-count min = 1, max = 5, default = 3
clid_authen_col_npw_cli.tcl	This script authenticates the account number and PIN respectively using ANI and null. If the authentication fails, it allows the caller to retry. The retry number is configured through the CLI. The account number and PIN are collected separately.	call application voice retry-count min = 1, max = 5, default = 3
clid_authen_collect_cli.tcl	This script authenticates the account number and PIN using ANI and DNIS. If the authentication fails, it allows the caller to retry. The retry number is configured through the CLI. The account number and PIN are collected separately.	call application voice retry-count min = 1, max = 5, default = 3
clid_col_npw_3_cli.tcl	This script authenticates using ANI and null for account and PIN respectively. If the authentication fails, it allows the caller to retry. The retry number is configured through the CLI.	call application voice retry-count min = 1, max = 5, default = 3
clid_col_npw_npw_cli.tcl	This script authenticates using ANI and null for account and pin respectively. If authentication fails, it allows the caller to retry. The retry number is configured through the CLI. The account number and PIN are collected together.	call application voice retry-count min = 1, max = 5, default = 3

Verifying IVR Configuration

You can verify IVR configuration by performing the following tasks:

- To verify IVR configuration parameters, use the **show running configuration** command.
- To display a list of all voice applications, use the **show call application summary** command.
- To show the contents of that script, use the **show call application voice** command.
- To verify that the operational status of the dial peer is up, use the **show dial-peer voice** command.

IVR Configuration Example

The following example shows the configuration for the IVR feature; this output was created by using the **show running configuration** command. The following IVR configuration uses the `clid_col_dnis_3.tcl` TCL IVR script.

```
Router # show running-config
Building configuration...

Current configuration:
!
! Last configuration change at 17:57:03 pst Wed Feb 24 1999
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname sblab115
!
boot system tftp username/c3620-is56i-mz 172.29.248.12
boot system tftp username/c3620-is-mz 172.29.248.12
no logging buffered
aaa new-model
aaa authentication login no_rad local
aaa authentication login h323 group radius local
!
clock timezone pst -8
clock summer-time pdt recurring
ip subnet-zero
ip domain-name cisco.com
ip name-server 172.29.248.16
ip name-server 171.69.187.13
!
call application voice c4 tftp://santa/username/clid_4digits_npw_3.tcl
call application voice c5 tftp://santa/username/clid_col_dnis_3.tcl
!
voice-port 1/0/0
!
voice-port 1/0/1
!
voice-port 1/1/0
shutdown
pre-dial-delay 0
!
voice-port 1/1/1
shutdown
```

```
pre-dial-delay 0
!
dial-peer voice 997 voip
 destination-pattern 997
 session target loopback:rtp
!
dial-peer voice 1 pots
 application clid
!
dial-peer voice 2 pots
!
dial-peer voice 100 pots
 application c5
 answer-address 1234
 destination-pattern 100
 port 1/0/0
!
voice-port 1/1/0
 shutdown
 pre-dial-delay 0
!
voice-port 1/1/1
 shutdown
 pre-dial-delay 0
!
dial-peer voice 997 voip
 destination-pattern 997
 session target loopback:rtp
!
dial-peer voice 1 pots
 application clid
!
dial-peer voice 2 pots
!
dial-peer voice 100 pots
 application c5
 answer-address 1234
 destination-pattern 100
 port 1/0/0
!
dial-peer voice 110 pots
 application clid
 destination-pattern 110
 direct-inward-dial
 port 1/1/0
!
dial-peer voice 111 pots
 destination-pattern 111
 port 1/1/1
!
dial-peer voice 114 voip
 destination-pattern 114...
 session target dns:sblab114
!
dial-peer voice 991 pots
 destination-pattern 991
 port 1/0/0
 session target loopback:uncompressed
!
dial-peer voice 992 pots
 destination-pattern 992
 port 1/0/1
 session target loopback:uncompressed
```