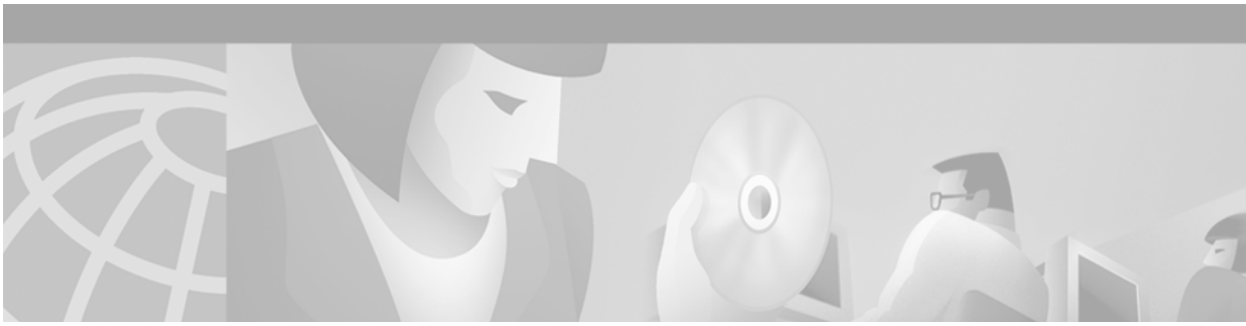


# Prepaid Distributed Calling Card via Packet Telephony

**Feature History**

| Release | Modification |
|---|---|
| 12.0(7)T | This feature was introduced. |
| 12.1(5)XM2 | Support was added for the Cisco AS5350 and Cisco AS5400 universal gateways. |
| CSCds35184 | Configuration requirement added to comply with DDTS CSCds35184. |

This feature is also known as “Debit Card for Packet Telephony on Cisco Access Platforms.”

The Debit Card feature gives service providers the ability to offer calling service with the use of debit accounting. This f eature was first introduced in Cisco IOS Release 12.0(4)XH. The following enhancements have been added to Cisco IOS Release 12.0(7)T:

- Support for the Spanish language
- New TCL scripts and script modifications
- Two new commands:
  - **ivr autoload**
  - **ivr prompt memory**

This document includes the following sections:


- Feature Overview, page 2
- Benefits, page 19
- Related Features and Technologies, page 19
- Supported Platforms, page 20
- Supported MIBs and RFCs, page 20
- Prerequisites, page 20
- Configuration Tasks, page 21
- Configuration Examples, page 25
- Command Reference, page 29

- Glossary, page 42

# Feature Overview

The Debit Card for Packet Telephony on Cisco Access Platforms is an application supported by the Cisco Interactive Voice Response (IVR) feature. The IVR voice scripts have been modified to use Tool Command Language (TCL) scripts.

**Note** For a complete description of the IVR TCL feature, see "Configuring Interactive Voice Response for Cisco Access Platforms." on the Cisco CCO Web site. http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv/5300/cfios/cfselfea/0061ivr.htm

The feature components consist of IVR functionality in Cisco IOS software that work in conjunction with an integrated third-party billing system. The Debit Card feature includes the ability to maintain per-user credit balance information through the use of a billing system. When these features are implemented, the billing system and IOS software functions enable a carrier to authorize voice calls and debit individual user accounts in real time at the edges of a voice-over-IP network, without requiring external service nodes. This feature uses vendor specific attributes (VSAs) to communicate with the billing system.

A general, the *Interactive Voice Response* software infrastructure allows pre-recorded audio files to be combined dynamically to play the dollar amount of credit remaining, the time and date, and other information. The dynamic creation of the audio files (by combining prerecorded audio pieces) is limited to auidio files playing out dollar, time, and day information.

We recommend that you contact your Cisco marketing or sales representative to help determine which RADIUS server vendor best meets your needs. Also, refer to the marketing data sheet titled "Debit Card Application for Cisco Packet Telephony" that describes the Debit Card feature and provides a list of the specific features supported by RADIUS server vendors.

## TCL Scripts

The Cisco Debit Card for Packet Telephony feature works in tandem with the Cisco Interactive Voice Response (IVR) feature modified with TCL scripts to provide voice prompts. These voice prompts are required to collect the necessary digits for accounting and settlement processes.

Before Cisco IOS Release 12.0(7)T and the introduction of IVR TCL scripts, IVR scripts were written in C programming language and compiled with the Cisco IOS software. This approaach tied the IVR application to the Cisco IOS software and also constrained releases of updated IVR scripts due to the Cisco IOS Release cycle.

Objects or commands written in C language that can be grouped together by the TCL script. When the script is activated by the incoming call, it activates the C codes in run-time mode and performs the work of these commands.

**Note** For additional information about the *Interactive Voice Response* application, see the IOS New Feature documentation "Configuring Interactive Voice Response for Cisco Access Platforms" at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv/5300/cfios/cfselfea/0061ivr.htm

The TCL scripts required for the Debit Card feature need to be downloaded from the Cisco CCO Software Center, located at the following URL:

http://www.cisco.com/cgi-bin/tablebuild.pl/tclware

- Cisco IOS Releases 12.0(7)T and 12.0XH1 are compatible with TCLware 1.0.x
- Cisco IOS Release 12.0(7)T is compatible with TCLware 1.1.x

## TCL Script Summaries

The TCL IVR scripts that are supported by the Debit Card feature in Cisco IOS Release 12.0(7)T are described in summary as follows:

| Script Name | Description |
|---|---|
| **clid_col_dnis_3.tcl** | Authenticates the caller ID three times, first with DNIS, if that is not successful, it attempts to authenticate with the caller PIN up to 3times. |
| **clid_col_npw_3.tcl** | Authenticates with NULL. If authentication is not successful it attempts to authenticate 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 fourteen digit account number and password entered together. |
| **clid_4digits_npw_3_cli.tcl** | 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 command line interface (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 command line interface (CLI). The account number and PIN are collected separately. |
| **clid_authen_collect_cli.tcl** | 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 command line interface (CLI). The account number and PIN are collected separately. |

| Script Name | Description |
|---|---|
| **clid_col_npw_3_cli.tcl** | 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 command line interface (CLI). |
| **clid_col_npw_npw_cli.tcl** | 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. |

Two new TCL scripts have been developed for Cisco IOS Release 12.0(7)T:

- **debitcard.tcl**—Collects account ID and PIN with a single prompt.
- **debitcard_acct_pin.tcl**—Requests the account and PIN number with two separate prompts.

# Debit Card Features

**Note** The feature functionality is dependent upon the working configuration of the designated RADIUS server. Different Debit Card features may be supported on different RADIUS servers that control the VSA attributes. It is the customer's responsibility to determine which RADIUS server vendor best meets the customer's needs.

New feature functionality includes:

- Rates a call according to the caller ID, PIN, and destination number.

  The call is authenticated using caller ID and PIN. The RADIUS server provides user credit (dollar) amount. The caller is then prompted to enter the destination number. The TCL script authorizes the call with the RADIUS server. The RADIUS server is expected to keep track of the caller's credit information, and it communicates with other billing servers, if necessary, to maintain or furnish user credit information.

  An enhancement was added to Cisco IOS Release 12.0(7)T to allow two separate prompts for requesting the caller account number and password.

- Plays the credit (dollar amount) remaining on a card in $$$$$$.$$ format.

  The RADIUS server is expected to maintain the credit information and furnish it to the script at the time of authentication. The TCL script plays a prompt that announces the remaining credit to the caller as dollars and cents. This is achieved by prerecording the words 'one', 'two', and so on, and then combining these recordings to form the final prompt. The format and resolution is $$$$$$.$$. The design is flexible enough to play any amount up to a maximum of 999999.99. For example, "You have one-thousand, three-hundred forty-five dollars and seventeen cents" is played.

- Announces the time remaining credit on the card in hours and minutes (HH:MM).

  The RADIUS server provides the credit (time) remaining to the script during authorization. The TCL script combines prerecorded audio files to form the final prompt and play the 'time-remaining' to the caller. The time credit amount returned by the RADIUS server is expected to have taken into

consideration the ‘rating’ and time boundary overlaps. The prompt played to the caller is, for example, “You have five hours and thirty-five minutes.” The design is flexible enough to play any amount of time up to a maximum time specified in the script.

If the time credit remaining is less than the configurable “time-out” value, the prompt played for example, “You only have two minutes remaining on your credit.”

- Plays a ‘time-running-out’ message based on the configurable time-out value.

  The RADIUS server maintains and furnishes time credit information during the authorization. The TCL script monitors the time remaining and, based on the configured value, plays a ‘time-has-run-out’ message to the caller. The called party hears silence during this time. For example, if the timeout value is configured for three minutes, the prompt, “You only have three minutes remaining on your credit” is played.

- Plays a warning ‘time-has-run-out’ message when the credit runs out.

  This message is played to the calling party by the TCL script when the time credit has run out. The called party hears silence. The message is, for example, ‘Sorry, you have run out of credit.’

- Makes more than one successive call to different destinations during a single call session.

  This feature makes it possible for the caller to make subsequent calls to different destinations without disconnecting from the call leg. Thus, the caller is required to enter the account ID number and PIN only once during initial authorization. To make subsequent calls, the caller needs to enter only the destination number. After talking with one destination, the caller is allowed to disconnect the call by pressing the pound (#) key on the keypad and holding it down for one to two seconds. If the # key is pressed down for more than one second it is treated as long pound (#). The called party is disconnected and an announcement is played to the caller, giving the new time, and prompting for a new destination number.

  This feature also allows the caller to make additional calls if the called party hangs up.

- Reauthorizes each new call.

  Every time a caller enters a new destination number, the TCL script reauthorizes the call with the RADIUS server and obtains the remaining time and credit balance information. The TCL script then announces the amount of time remaining to the calling party.

- Allows type-ahead keypad entries without waiting for the prompt to complete.

  The normal terminating character for the caller ID, PIN, and destination number is the # key. The caller may want to continue without waiting to hear the prompts. This TCL script will stop playing or will not start a prompt when it discovers that the caller wants to type ahead.

- Allows the caller to skip past announcements by pressing a touch-tone key.

  This TCL script stops playing announcements when the system determines that the caller has pressed any touch tone key.

- Allows retry when entering data (user-ID/PIN/destination number) by using a special key.
  - The caller is allowed to interrupt partially entered numbers and restart from the beginning by pressing asterisk (*) key on the keypad. The asterisk key is configured in the TCL script. The caller can use the asterisk key to cancel an entry and reenter the user-ID/PIN/destination number. The caller is allowed to reenter data only a certain number of times. The number of retries is configurable; the default is three.

- Terminates a field by size rather than using terminating character (#).

The TCL script can be used to specify a number of digits in the user ID and PIN fields. This means that the caller can type all of the digits (without the terminating character) and the script determines how to extract different fields from the number strings. If the caller uses the terminating character (the pound key), the terminating character takes precedence and the fields are extracted accordingly.

- Supports two languages.

  The language is selected when the caller presses a predefined key. For example, 'For English press one. For Mandarin press two.' The TCL script uses the selected language until the caller disconnects, the caller is asked only once, at the beginning of the session, for the language of choice.

  Support for the Spanish language is introduced with Cisco IOS Release 12.0(7)T.

  The **ivr autoload** Cisco IOS command is used during configuration to determine which set of audio files to load when the gateway server boots up.

- Sends an off-net tone to the caller.

  The RADIUS server is expected to maintain information regarding off-net calls. During authorization, it provides this information to the TCL script. Based on the collected information, the TCL script has the ability to generate a prerecorded message or tone to the calling party.

- Supports vendor specific attributes.

  Cisco vendor specific attributes (VSAs), which before this release were overloaded into the Session ID field of the AAA application.

- Provides voice-quality information to the RADIUS server on a call-by-call basis.

  A new field has been added to the Stop Record field. The data for this field is obtained from fields that maintain and tune voice quality. It is up to the user application on the RADIUS server to use this information and give credit to the caller if the call has unsatisfactory voice quality.

- Uses prompt memory more efficiently.

  When voice prompts are not used for a period of time, they are swapped out of RAM. The swapping does not introduce undue delays in playing prompts. The most frequently used prompts remain in memory and are not swapped.

- Creates dynamic prompts by using prerecorded audio files.

  A general infrastructure is provided that allows combining prerecorded audio files to play the dollar amount , time, and day. An interface for the scripts to use this infrastructure is also part of this feature. Dynamic creation of the final audio (by combining prerecorded audio files) is limited to playing out dollar amount, time, and day information. For example, when the system receives a credit balance of $15.50, it combines the prerecorded audio files, "You have" "fifteen" "dollars" "and" "fifty" "cents" to make up this message.

- Allows retries for RADIUS server failures, with the maximum number retries allowed determined by the RADIUS server.

  If errors that are returned by the RADIUS server during authentication or authorization (use of the AAA application), the caller is allowed to retry the entry. The RADIUS server determines how many retries to allow. The caller is disconnected when the number of retries has exceeded the limit.

- Determines how many languages are configured and only plays the language selection menu if needed.

## VSA Fields

VSAs are collected by the RADIUS server during the accounting process when AAA is configured with the Debit Card feature. (See "Configuring AAA" on page 22.) Data items are collected for each call leg created on the gateway. A call leg is the internal representation of a connection on the gateway. Each call made through the gateway consists of two call legs: incoming and outgoing. The call leg information emitted by the gateways can be correlated by the connection ID, which is the same for all call legs of a connection.

Use the H.323 VSA method of accounting when configuring the AAA application. (See the command "gw-accounting" on page 37.)

There are two modes:

- Overloaded Session-ID
  Use the **gw-accounting h323 syslog** command to configure this mode.
- VSA
  Use the **gw-accounting h323 vsa** command to configure this mode.

### Overloaded Session-ID Fields

The voice specific accounting information is coded into the 'session-id' field of the RADIUS record. Each field is encoded as an ASCII string. The fields are separated by a /. For additional information about the overloaded session ID refer *Service Provider Features for Voice over IP. (*See "Related Documents" on page 19.) Table 1 shows the VSA fields and a brief description of each.

***Table 1 VSA Attributes in Overloaded Session ID***

| Field | Description |
|---|---|
| *session id* | The standard RADIUS account session ID. |
| *setup time* | The Q.931 setup time for this connection in NTP format. |
| *gateway id* | The name of the underlying gateway in the form of "gateway.domain_name". |
| *call origin* | Origin of the call relative to the gateway. Possible values are **originate** and **answer**. |
| *call type* | Indicates call leg type. Possible values are **telephony** and **VoIP**. |
| *connection id* | A unique global identifier used to correlate call legs that belong to the same end-to end call. The field consists of 4 long words (128 bits). Each long word is displayed as a hexadecimal value and separated by a space character. |
| *connect time* | The Q.931 connect time for this call leg, in NTP format. |
| *disconnect time* | The Q.931 disconnect time for this call leg, in NTP format. |
| *disconnect cause* | Documented in the Q.931 specification. Can be in the range of 1 to 160. |
| *remote IP address*[1] | Address of the remote gateway port where the call is connected. |

1. Support for the **remote IP address** field was introduced with Cisco IOS Release 11.3(7)NA.

### VSA Fields

The voice specific accounting information is coded as separate VSAs. Each attribute is encoded as 'Cisco-AV-pair' which has the format attribute = value. The value is encoded as an ASCII string.

The VSA fields and their ASCII values listed in Table 2.

*Table 2 VSA Fields and Their ASCII Values*

| Field | Attribute Name | Value |
|---|---|---|
| gw_id | h323-gw-id | Gateway name |
| conf_id | h323-conf-id | Conference ID |
| call_origin | h323-call-origin | Telephony/VOIP |
| call_type | h323-call-type | Originating/terminating |
| setup_time | h323-setup-time | Setup time in UTC (GMT) |
| connect_time | h323-connect-time | Connect time in UTC |
| disconnect_time | h323-disconnect-time | Disconnect time in UTC |
| disconnect_cause | h323-disconnect-cause | Cause code (reference Q.931) |
| remote_address | h323-remote-address | IP address of remote gateway |
| voice_quality | h323-voice-quality | Quality based on ICPIF |

## Debit Card Call Flow

A high-level call flow sequence is displayed in Figure 1 through Figure 5 “Debit Card Call Sequence”. The actual call flow varies, depending on the parameters passed to the application and also on the features that are available on the RADIUS server billing system that is being used.

The call sequence flow charts below graphically depict the different states in the Debit Card application. The different states are represented by the double bars in the boxes and show the flow from one state to the next.

*Figure 1 Debit Card Call Sequence*

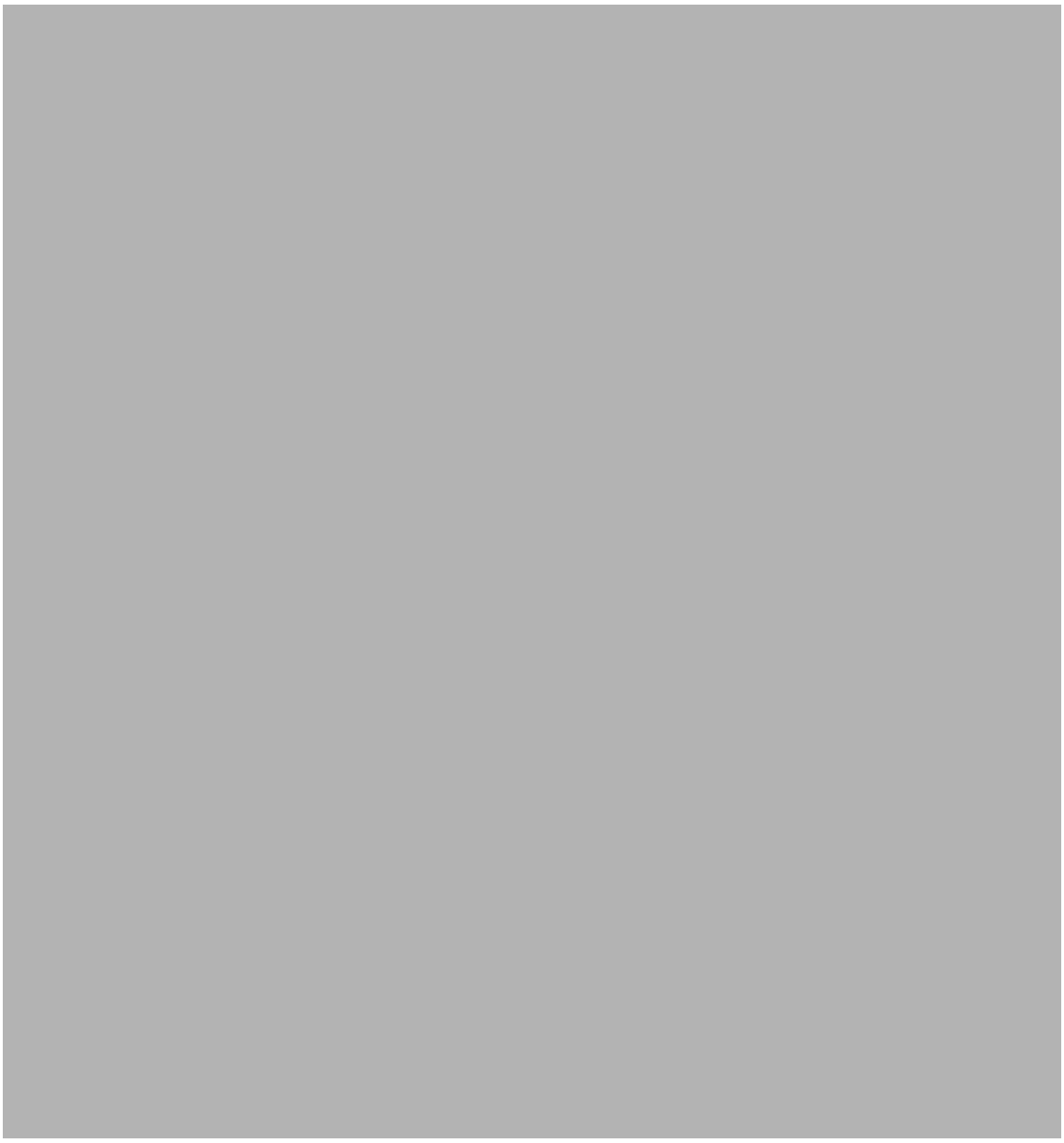


*Figure 2*

***Figure 3***

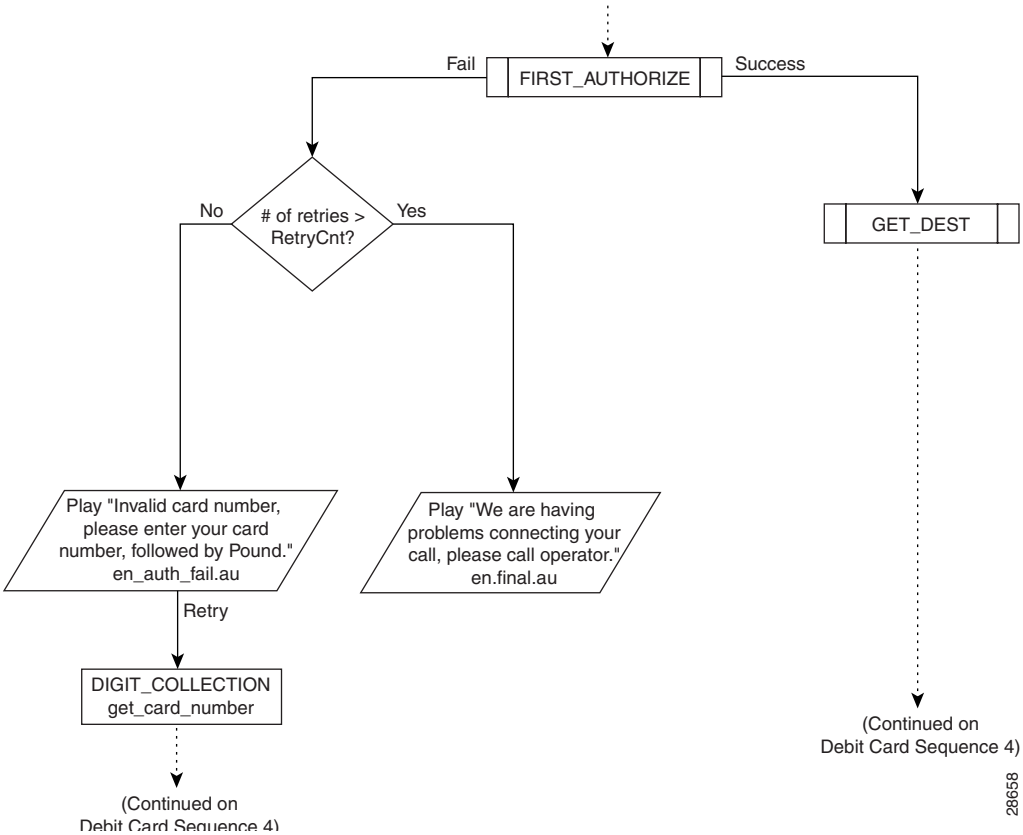

Fail
FIRST_AUTHORIZE
Success
No
# of retries > RetryCnt?
Yes
GET_DEST
Play "Invalid card number, please enter your card number, followed by Pound." en_auth_fail.au
Play "We are having problems connecting your call, please call operator." en.final.au
Retry
DIGIT_COLLECTION get_card_number
(Continued on Debit Card Sequence 4)
(Continued on Debit Card Sequence 4)
28658

***Figure 4 h***

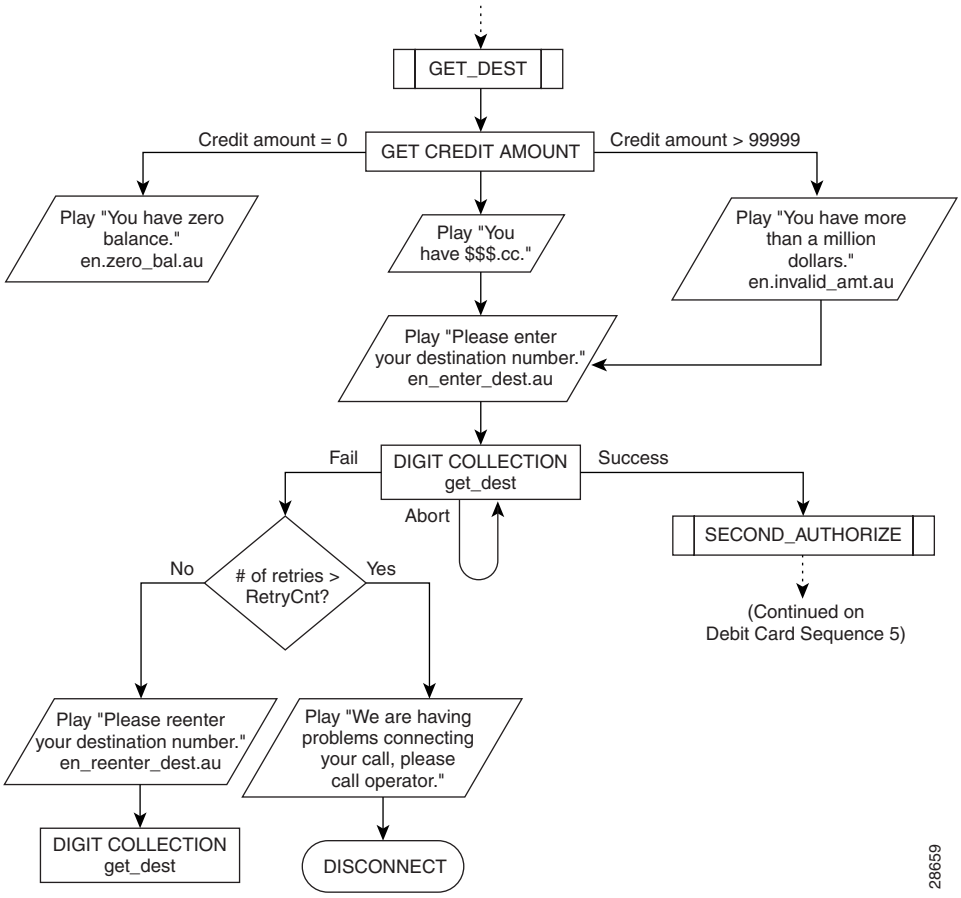
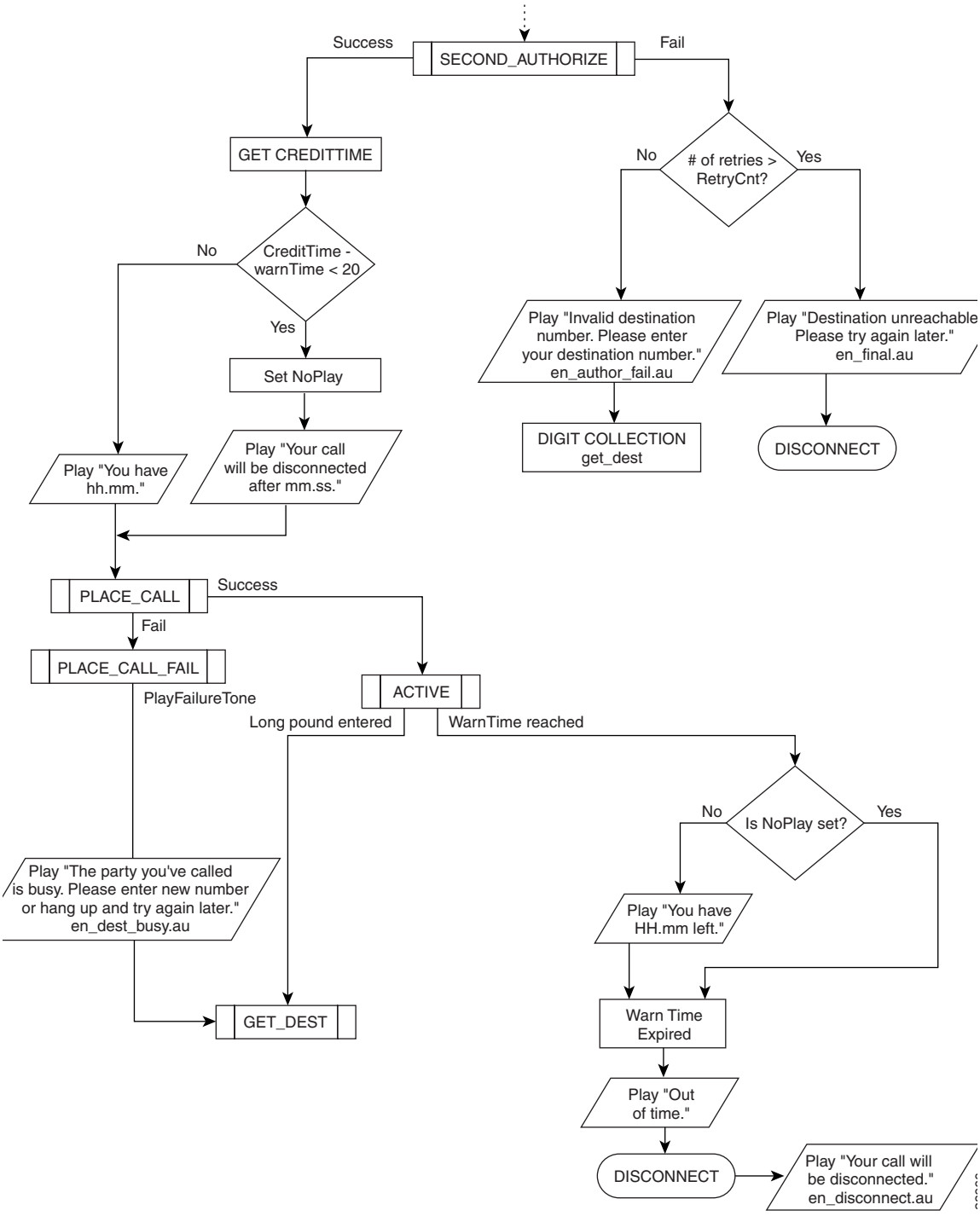

GET_DEST
Credit amount = 0
GET CREDIT AMOUNT
Credit amount > 99999
Play "You have zero balance." en.zero_bal.au
Play "You have $$$.cc."
Play "You have more than a million dollars." en.invalid_amt.au
Play "Please enter your destination number." en_enter_dest.au
Fail
DIGIT COLLECTION get_dest
Success
Abort
SECOND_AUTHORIZE
No
# of retries > RetryCnt?
Yes
(Continued on Debit Card Sequence 5)
Play "Please reenter your destination number." en_reenter_dest.au
Play "We are having problems connecting your call, please call operator."
DIGIT COLLECTION get_dest
DISCONNECT
28659

***Figure 5***


SECOND_AUTHORIZE
Success
Fail
GET CREDITTIME
# of retries > RetryCnt?
No
Yes
CreditTime - warnTime < 20
No
Yes
Play "Invalid destination number. Please enter your destination number." en_author_fail.au
Play "Destination unreachable Please try again later." en_final.au
Set NoPlay
DIGIT COLLECTION get_dest
DISCONNECT
Play "You have hh.mm."
Play "Your call will be disconnected after mm.ss."
PLACE_CALL
Success
Fail
PLACE_CALL_FAIL
ACTIVE
PlayFailureTone
Long pound entered
WarnTime reached
No
Is NoPlay set?
Yes
Play "The party you've called is busy. Please enter new number or hang up and try again later." en_dest_busy.au
Play "You have HH.mm left."
GET_DEST
Warn Time Expired
Play "Out of time."
DISCONNECT
Play "Your call will be disconnected." en_disconnect.au

# Audio File Prompts

Cisco provides a set of professionally recorded English (U.S.) and Mandarin audio prompts to allow easier immediate integration and use of the Debit Card feature. The prompts are stored on the FTP server in the audio file format (.au.) To announce the credit available to the caller, the system concatenates a series of prompts to create the appropriate message.

The Cisco provided audio files are zipped and stored on CCO in the URL for the Software Support Center. The audio files need to be downloaded to either a TFTP server or into Flash memory. When the system runs, these files are copied into memory.

**Note** You can find TCLWare and audio files at the following URL: http://www.cisco.com/cgi-bin/tablebuild.pl/tclware

Because there are a large number of prompts, and to ensure efficient use of system memory resources, all prompts are downloaded to a TFTP server. A basic set of audio files is downloaded to the system when it is initialized. The system removes less-frequently-used prompts from memory to conserve memory: when the prompt is needed, the system fetches the prompt from the TFTP server.

For languages that are syntactically similar to English, the audio file can be recorded and saved as the same filename to allow the system to construct the message properly. (See Audio File Naming Convention, page 18.) It is not mandatory to use the prompt set that Cisco provides. If the customer has access to a recording studio, prompts can be created or customized, as long as they are saved in the proper format.

## Cisco Provided Audio Files

The following audio file prompts are provided by Cisco. A similar set is available in Mandarin. Not all of the audio file sets are used by Cisco IOS Release 12.0(7)T. Some are reserved for future use.

Support for the Spanish language is added to Cisco IOS Release 12.0(7)T. A similar set of professionally recorded audio files is available on CCO. The audio files provided are listed in Table 3 and Table 4.

*Table 3 Numbers Audio File Set*

| Audio File Name | Recorded Prompt | Audio File Name | Recorded Prompt |
|---|---|---|---|
| en_zero.au | Zero | en_fifteen.au | Fifteen |
| en_one.au | One | en_sixteen.au | Sixteen |
| en_two.au | Two | en_seventeen.au | Seventeen |
| en_three.au | Three | en_eighteen.au | Eighteen |
| en_four.au | Four | en_nineteen.au | Nineteen |
| en_five.au | Five | en_twenty.au | Twenty |
| en_six.au | Six | en_thirty.au | Thirty |
| en_seven.au | Seven | en_forty.au | Forty |
| en_eight.au | Eight | en_fifty.au | Fifty |
| en_nine.au | Nine | en_sixty.au | Sixty |

*Table 3 Numbers Audio File Set (continued)*

| Audio File Name | Recorded Prompt | Audio File Name | Recorded Prompt |
|---|---|---|---|
| en_ten.au | Ten | en_seventy.au | Seventy |
| en_eleven.au | Eleven | en_eighty.au | Eighty |
| en_twelve.au | Twelve | en_ninety.au | Ninety |
| en_thirteen.au | Thirteen | en_hundred.au | Hundred |
| en_fourteen.au | Fourteen | en_thousand.au | Thousand |

*Table 4 Miscellaneous Prompts*

| Audio File Name | Recorded Prompt |
|---|---|
| en_second.au | Second |
| en_seconds.au | Seconds |
| en_minute | Minute |
| en_minutes | Minutes |
| en_hour.au | Hour |
| en_hours.au | Hours |
| en_cent.au | Cent |
| en_cents.au | Cents |
| en_dollar.au | Dollar |
| en_dollars.au | Dollars |

**Additional Miscellaneous Prompts:**

- **en_welcome.au**

  “Welcome to Cisco Debit Card Demo”.
- **en_lang_select.au**

  “Please press 1 for English, 2 for Mandarin.”
- **en_wrong_lang_sel.au**

  “You have made an invalid selection. Please press 1 for English or press 2 for Mandarin.”
- **en_no_lang_sel.au**

  “You did not select any language. Press 1 for English or press 2 for Mandarin.”
- **en_final.au**

  “We are having difficulties connecting your call. Please try again later.”
- **en_generic_final.au**

  “Please hang up and try again.”
- **en_enter_card_num.au**

  “Please enter card number followed by pound.”

- **en_invalid_digits.au**

  "You have entered an invalid number of digits. Please re-enter your card number followed by pound."

- **en_auth_fail.au**

  "You have entered an invalid card number. Please re-enter your card number followed by pound."

- **en_no_card_entered.au**

  "You did not enter any digits. Please enter card number followed by pound."

- **en_technical_problem.au**

  "We are having technical difficulties. Please call back later."

- **en_zero_bal.au**

  "You have zero balance. Please call the operator or hang up."

- **en_enter_dest.au**

  "Please enter destination number."

- **en_disconnect.au**

  "Your call will be disconnected."

- **en_disconnected.au**

  "You have been disconnected."

- **en_dest_collect_fail.au**

  "Sorry, the number you have dialed is blocked. If you feel you have reached a number in error, please call the customer service number."

- **en_invalid_amt.au**

  "You have more than one million."

- **en_dest_busy.au**

  "The party you called is busy, please enter a new number or hang up and try again later."

- **en_enter_acct.au**

  "Please enter your account number followed by the pound key."

- **en_no_acct_entered.au**

  "We did not get any input, please enter your account number followed by the pound key."

- **en_invalid_digits_acct.au**

  "You have entered an invalid number of digits. Please enter your account number followed by the pound key."

- **en_invalid_account.au**

  "You have entered an invalid account number. Please enter your account number followed by the pound key."

- **en_enter_pin.au**

  "Please enter your PIN number followed by the pound key."

- **en_no_pin_entered.au**

  "We did not get any input, please enter your pin number followed by the pound key."

- **en_invalid_digits_pin.au**

  "You have entered an invalid number of digits. Please enter your PIN number followed by the pound key."
- **en_invalid_pin.au**

  "You have entered an invalid PIN. Please enter your pin number followed by the pound key."
- **en_card_expired.au**

  "We are sorry, your card has expired."
- **en_account_blocked.au**

  "This account is currently in use."
- **en_no_dest_entered.au**

  "We did not get any input. Please enter the destination number you are calling."
- **en_invalid_digits_pin.au**

  "You have entered an invalid number of digits. Please enter your PIN number followed by the pound key."
- **en_invalid_pin.au**

  "You have entered an invalid PIN. Please enter your PIN number followedby the pound key."
- **en_card_expired.au**

  "We are sorry, your card has expired."
- **en_account_blocked.au**

  "This account is currently in use."
- **en_no_dest_entered.au**

  "We did not get any input. Please enter the destination number you are calling."
- **en_no_dialpeer_match.au**

  "You have entered an invalid destination. Please re-enter the destination number you are calling."
- **en_connect_cust_ser.au**

  "You will be connected to Customer Service."
- **en_dial_cust_ser.au**

  "Please hang up and dial the calling card customer service number."
- **en_no_service.au**

  "We are sorry, this service is not avaliable."
- **en_dest_unreachable.au**

  "We are sorry, the destination you have called is unreachable."
- **en_toll_free.au**

  "You can only make toll-free calls."

## Audio File Naming Convention

If you record your own audio files, you must name them using the convention described in Table 4 for the TCL scripts to identify which audio file to use. The TCL scripts are designed to work with designated audio file names.

For example, when the Debit Card audio file for the caller to choose a language (**en_lang_select.au**) is played, ("Please press # 1 for English and # 2 for Mandarin"), if Mandarin is selected by the caller, then the TCL script calls the <ch> audio files to interact with the system.

Continuing with this example, when configuring the voice platform to process Mandarin audio files, use the **call application voice** command, with the **language** and *set-location* key words. When specifying the *set-location* parameter, the correct <language> <ch> specifier needs to be configured to interact with the TCL script. Therefore, when naming your audio files, make sure you include the language identifier for each file.

***Table 5 Example Audio File Naming Convention***

| Audio File Name | Description |
|---|---|
| en_one.au | Specifies the English language audio file for the number one. |
| ch_one.au | Specifies the Mandarin language audio file for the number one |
| sp_one.au | Specifies the Spanish language audio file for the number one. |

**Note** See "Command Reference" on page 29 and the command "call application voice application_name location".

## Creating Audio Index Files

If you record your own audio files, you must also create an index file that contains a list of the audio files in URL format. An index file needs to be created for each audio file that needs to be downloaded from TFTP to memory. Use the **ivr autoload** command to download the audio files into Flash memory. See"Command Reference" on page 29 for additional command description and instructions for downloading using the *index* parameter.

When creating your audio file index, keep in mind that the filename and extension (au) are is actually the file's URL. Follow these recommendations:

- Each line should list only one file location (URL)
- Comment lines start with #
- Locations listed in the index file should match the locations used by the application (see the **call application voice set-location** command)
- Extra white spaces at the beginning and end of the line are ignored
- No white spaces are allowed within the URL

### Sample Index File

The following is a sample index file:

```
# tftp://jurai/tclware/au/en/auth_fail_final.au
# tftp://jurai/tclware/au/en/auth_fail_retry.au

# tftp://jurai/tclware/au/en/auth_fail_retry_number.au
# tftp://jurai/tclware/au/en/auth_failed.au

# tftp://jurai/tclware/au/en/ch_generic_final.au
# tftp://jurai/tclware/au/en/ch_lang_sel1.au
```

# Benefits

- The Cisco Debit Card application makes deployment of a prepaid calling card economically feasible in a network that includes a small point of presence (POP).
- The addition of TCL, which is a programmable scripting language, allows the separation of the Cisco IOS software and the IVR feature software. This allows quicker turnaround for additional TCL scripts and a vast reduction in memory requirements.

## Restrictions

The maximum call length is 18 hours. When a caller uses the Debit Card application, a call is disconnected after 18 hours, even if the caller has a dollar amount balance remaining. This limitation is due to the timing implementation in the software.

# Related Features and Technologies

- “Service Provider Features for Voice over IP”
- “Interactive Voice Response for Cisco Access Platforms”
- “Named Method Lists for AAA Authorization, Authentication and Accounting”

## Related Documents

For related information on this feature, see the following documents:

- *Service Provider Features for Voice over IP* (introduced in Cisco IOS Release 12.0(3)T)
- *Interactive Voice Response Feature for Cisco Access Platforms* (introduced in Cisco IOS Release 12.0(7)T)
- Voice over IP for the Cisco AS5300
- Configuring Voice over IP for the Cisco 2600 and Cisco 3600 Series Routers

# Supported Platforms

This feature is supported on the following platforms:

- Cisco 2600 series routers
- Cisco 3600 series routers
- Cisco AS5300 universal access server
- Cisco AS5350 universal gateway
- Cisco AS5400 universal gateway

# Supported MIBs and RFCs

### Standards

This feature adds support for the following ITU-T standards: H.323 Annex E and H.323 Annex G.

### MIBs

No new or modified MIBs are supported by this feature.

To obtain lists of MIBs supported by platform and Cisco IOS release and to download MIB modules, go to the Cisco MIB web site on Cisco Connection Online (CCO) at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

### RFCs

The RADIUS RFC is already implemented in the Cisco IOS software. Extensions have been added that are compliant with the following IETF RFCs:

- 2138
- 2139

# Prerequisites

The Cisco AS5350 and Cisco AS5400 do not support the Mica Modem Card, Microcom Modem Card, or VoIP Feature Card. Voice and modem functions are provided by the Universal Port Dial Feature card running SPE firmware. See the *Cisco AS5350 Universal Gateway Card Installation Guide* and the *Cisco AS5400 Universal Gateway Card Installation Guide* for more information. All references to the Cisco AS5300 in this document apply to the Cisco AS5350 and Cisco AS5400 platforms with the following exceptions:

- Use the Universal Port Dial Feature Card instead of the Mica or Microcom modem cards.
- Use SPE firmware instead of portware version 6.7.7.
- Run Cisco IOS Release 12.1(5)XM2 software for VoIP functionality.

### Other Prerequisites

Before you can configure your access server platform (Cisco AS5300, Cisco 3600, or other supported voice platform) with the Debit Card feature, complete the following activities:

- Ensure that your access platform has a minimum of 16 MB Flash memory and 64 MB DRAM memory.

- Establish a working IP network. For more information about configuring IP, refer to the "IP Overview," "Configuring IP Addressing," and "Configuring IP Services" chapters in the Cisco IOS Release 12.0 *Network Protocols Configuration Guide, Part 1*.
- Configure Voice-over-IP. For more information about configuring Voice-over-IP, refer to the *Voice over IP Software Configuration Guide* for the appropriate access platform.
- Configure a TFTP server to perform storage and retrieval of the audio files which are required by the Debit Card gateway.
- Program and configure the interface between the RADIUS billing server and the Cisco gateway to operate with VSAs.
- Ensure the correct software is loaded on the platform that supports the gateway feature and supporting software applications. Cisco IOS Release 12.0(7)T or higher, and VCWare version 4.0.2 is required on the Cisco AS5300 universal access server.
- Download the TCL scripts required for the Debit Card feature, located at the following URL:

  http://www.cisco.com/cgi-bin/tablebuild.pl/tclware

  - Cisco IOS Releases 12.0(7)T and 12.0XH1 are compatible with TCLware 1.0.x
  - Cisco IOS Release 12.0(7)T is compatible with TCLware 1.1.x

# Configuration Tasks

Before starting the software configuration tasks for the Debit Card feature, complete the following preinstallation tasks:

- Download the TCL Scripts and audio files to be used with the Debit Card feature from the CCO Software Support Center.
- Store the TCL scripts and audio files on a TFTP server configured to interact with your gateway access server.

  **Note** If you have recorded your own audio files, make sure your files follow the correct naming convention (see "Audio File Naming Convention") and your audio file index meets the format requirements (see Creating Audio Index Files).

- Create an application to interact with the TCL scripts.

  Use the **call application voice** app_*name* set-*location* command to create the application. Enter any name that is significant to how you use the feature and then enter the location where the TCL scripts are located. (See the "call application voice application_name location" section on page 30.

- Pass the parameters to the application that you created using the **call application voice** command and the keywords {*language* | *redirect-number* | *pin-length* | *retry-count* | *set-location* | *uid-lenth* | *warning-time*}. See "call application voice application_name location" section on page 30.

**Note** When configuring the language priority using the **call application voice language** command, keep in mind that the software is hardcoded for configuring the digit 1 to represent the primary language, and the digit 2 to represent the secondary language.

When these tasks have been completed, you can begin the software configuration to configure the Cisco platform to perform Debit Card Calling services.

- Associate the call application you created to a dial-peer by configuring the inbound POTS dial peer. (See Configuring the Inbound POTS Dial Peer.)
- Enable VoIP gateway accounting by configuring AAA in the inbound dial peer. (See Configuring AAA, page 22.)

**Note** When configuring a voice port, use the following configuration designations:
For the Cisco AS5300 access server, port designation is *port*.
For the Cisco AS5350 and Cisco AS5400 gateway, port designation is *slot/port*.
For the Cisco AS5800 access server, port designation is *shelf/slot/port*.

# Configuring the Inbound POTS Dial Peer

To configure the inbound POTS dial peer, use the following commands:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `Router# configure terminal` | Enter global configuration mode. |
| **Step 2** | `Router(config)# dial-peer voice` *number* `pots` | Enter dial-peer configuration mode to configure a POTS dial peer.<br>**Note** The *number* value of the **dial-peer voice pots** command is a tag that uniquely identifies the dial peer. |
| **Step 3** | `Router(dial-peer)# application` *name* | Enter the selected TCL application name. |

# Configuring AAA

The Cisco IOS software AAA accounting user interface can be configured to use the H.323 method as follows:

The **authentication** command line creates a method list named H.323 with RADIUS being its only member.

Also note that the accounting command line looks like a regular RADIUS accounting command line for connection accounting. Connection accounting has to be globally enabled using this command line. Start-stop or stop only methods may be used.

| | Command | Purpose |
|---|---|---|
| **Step 1** | `Router#` **`enable`**<br><br>`Password: <password>`<br><br>`Router#` | Enter enable mode.<br><br>Enter the password. |
| **Step 2** | `Router#` **`config term`**<br>`Enter configuration commands, one per line. End with CNTL/Z.`<br>`Router(config)#` | Enter global configuration mode. You have entered global configuration mode when the prompt changes to `router(config)#`. |
| **Step 3** | `Router(config)#` **`aaa new-model`** | Initiates the AAA script. |
| **Step 4** | `Router(config)#` **`aaa authentication login h323 radius`** | Configure the router to use the H.323 method list for authentication purposes. |
| **Step 5** | `Router(config)#` **`aaa accounting connection h323 start-stop radius`** | Tell the system to use connection based accounting and the H.323 service. |
| **Step 6** | `Router(config)#` **`radius-server host 171.69.58.104 auth-port 1812 acct-port 1813`** | Sets the server host IP address and the ports for both the authentication service and the accounting service. |
| **Step 7** | `Router(config)#` **`radius-server key testing123`** | Specify the password used between the gateway and the RADIUS server. |
| **Step 8** | `Router(config)#` **`end`** | End the configuration session. |

**Note** When using the Debit Card Cisco IOS feature you must use the H.323 VSA method of accounting.

### Tips

- Because Cisco security authenticates based on account number, RADIUS is required for the redialer fax application.
- RADIUS is turned on globally, but is only used for services if it is so programmed. Fax hop-on does use it, and a regular session application does not.

## Verifying the Debit Card Configuration

Execute the **show call application voice summary** command to verify that the newly created application is listed.

```
Router #show call app voice summ
name                 description

session              Basic app to do DID, or supply dialtone.
fax_hop_on           Script to talk to a fax redialer
clid_authen          Authenticate with (ani, dnis)
clid_authen_collect  Authenticate with (ani, dnis), collect if that fails
clid_authen_npw      Authenticate with (ani, NULL)
clid_authen_col_npw  Authenticate with (ani, NULL), collect if that fails
clid_col_npw_3       Authenticate with (ani, NULL), and 3 tries collecting
clid_col_npw_npw     Authenticate with (ani, NULL) and 3 tries without pw

prepaid              tftp://keyer/debitcard.tcl

Router #

Next, execute the show dial-peer voice <peer tag> command and verify that the application
associated with the dial-peer is correct.
Router #show dial-peer voice 30001
VoiceEncapPeer30001
        information type = voice,
        tag = 30001, destination-pattern = `300..',
        answer-address = `', preference=0,
        group = 30001, Admin state is up, Operation state is up,
        incoming called-number = `', connections/maximum = 0/unlimited,
        application associated:prepaid
        type = pots, prefix = `300',
        session-target = `', voice-port = `2:D',
        direct-inward-dial = disabled,
        register E.164 number with GK = TRUE
        Connect Time = 0, Charged Units = 0,
        Successful Calls = 0, Failed Calls = 20,
        Accepted Calls = 20, Refused Calls = 0,
        Last Disconnect Cause is "4F  ",
        Last Disconnect Text is "service not implemented",
        Last Setup Time = 24591
Router #
```

# Configuration Examples

## Debit Card Feature Configuration Example

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

Current configuration:
!
version 12.0
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
service internal
!
hostname Router name
!
no logging buffered

! AAA configuration
!-------------------------------------
aaa new-model
aaa authentication login h323 group radius
aaa authorization exec h323 group radius
aaa accounting connection h323 start-stop group radius
!-------------------------------------
!
enable secret 5 $1$rLpz$DpgRh8qfaDqCPteN4/KXD0
enable password xxx
!
username Router password 0 xyxyxy
username s
!
!
resource-pool disable
!
!
!
!
!
!
ip subnet-zero
no ip domain-lookup

!  TFTP address configuration
!----------------------------------
ip host keyer 223.255.254.254


! prepaid application creation
!----------------------------------
call application voice prepaid tftp://keyer/debitcard.tcl

! passing parameters to prepaid application
!----------------------------------
call application voice prepaid uid-len 4
call application voice prepaid language 1 en
call application voice prepaid language 2 ch
call application voice prepaid set-location en 0 tftp://keyer/
```

```
mta receive maximum-recipients 1024
!
dial-control-mib max-size 300
!
controller T1 0
 shutdown
 framing esf
 linecode b8zs
 cablelength short 133
!
controller T1 1
 shutdown
 framing esf
 linecode b8zs
 cablelength short 133
!
controller T1 2
 framing esf
 clock source line primary
 linecode b8zs
 cablelength short 133
 pri-group timeslots 1-24
!
controller T1 3
 framing esf
 clock source line secondary 1
 linecode b8zs
 cablelength short 133
 pri-group timeslots 1-24
!
!
voice-port 2:D
 timeouts call-disconnect 0
!
voice-port 3:D
 timeouts call-disconnect 0

! configuring voip gw accounting
!-------------------------------
gw-accounting h323 vsa

! associating application to dial-peer
!-------------------------------------
dial-peer voice 30001 pots
 application prepaid
 destination-pattern 300..
 port 2:D
 prefix 300
!
dial-peer voice 40001 pots
 destination-pattern 400..
 direct-inward-dial
 port 3:D
 prefix 400
!
dial-peer voice 50001 voip
 destination-pattern 500..
 session target ipv4:147.14.25.1
!
dial-peer voice 60001 voip
 destination-pattern 600..
 session target ipv4:147.14.25.1
!
process-max-time 200
```

```
!
interface Ethernet0
 description ip address 132.132.1.2 255.255.255.0
 ip address 1.13.103.1 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 load-interval 30
 no keepalive
 no cdp enable
!
interface Serial2:23
 description D-Channel - To Abacus
 no ip address
 no ip directed-broadcast
 isdn switch-type primary-5ess
 isdn protocol-emulate user
 isdn incoming-voice modem
 fair-queue 64 256 0
 no cdp enable
!
interface Serial3:23
 description D-Channel - To Abacus
 no ip address
 no ip directed-broadcast
 isdn switch-type primary-5ess
 isdn protocol-emulate user
 isdn incoming-voice modem
 fair-queue 64 256 0
 no cdp enable
!
interface FastEthernet0
 ip address 147.14.25.100 255.255.0.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 duplex full
 no cdp enable
 hold-queue 2048 in
!
interface Async1
 ip address 2.2.2.1 255.255.255.0
 no ip directed-broadcast
 encapsulation ppp
 shutdown
 async mode dedicated
 ppp authentication chap
 hold-queue 10 in
!
interface Group-Async1
 physical-layer async
 ip unnumbered Serial2:22
 no ip directed-broadcast
 encapsulation ppp
 no ip mroute-cache
 dialer in-band
 dialer idle-timeout 200000
 async default routing
 async mode interactive
 no peer default ip address
 no fair-queue
 no cdp enable
 hold-queue 10 in
!
```

```
router igrp 200
 network 1.0.0.0
 network 133.133.0.0
!
router igrp 300
 network 132.132.0.0
 network 133.133.0.0
 network 147.14.0.0
!
no ip http server
no ip classless
!
ip route 1.13.80.100 255.255.255.255 1.13.0.1
ip route 223.255.254.254 255.255.255.255 Ethernet0
!
!
logging history size 500

! configuring radius parameters
!---------------------------------------
radius-server host 1.13.80.100 auth-port 1812 acct-port 1813
radius-server key cisco
radius-server vsa send accounting
radius-server vsa send authentication

!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 exec-timeout 0 0
 password lab

! configuring the NTP
!------------------------------
ntp master 15

!
end
!
Router #
```

# Command Reference

This section documents the following new or modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.1 command reference publications. The commands preceded with an asterisk first appeared in Cisco IOS Release 12.0(3)T.

- **call application voice application_name location**
- **call application voice app_name param-name param-value**
- **call application voice app-name set-location**
- **gw-accounting**
- *** ivr autoload**
- *** ivr prompt memory**

# call application voice *application_name location*

To create an application and to indicate the location where the corresponding TCL files, that implement this application are located, use the **call application voice** *application_name location* command.

**call application voice** *app_name location*

## Syntax Description

| | |
|---|---|
| *application_name* | A short string of characters that is the application name. |
| *location* | The location of the TCL file in URL format. Valid storage locations are TFTP, FTP, and Flash. |

## Defaults

No default behavior or values.

## Command Modes

Global configuration.

## Command History

| Release | Modification |
|---|---|
| 12.0(7)T | This command was introduced. |
| 12.1(5)XM2 | The command was introduced for the Cisco AS5350 and CiscoAS5400. |

## Usage Guidelines

This command is used with the Debit Card feature.

## Examples

```
call application voice prepaid tftp://keyer/debitcard.tcl
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice** *application_ name parameter-name parameter-value* | Passes parameters to the application created. |
| **call application voice set-location** | Indicates where the audio files are stored. |

# call application voice *app_name param-name param-value*

To pass parameters to the application created, use the **call application voice** *application_ name parameter-name parameter-value* command . This command is used in conjunction with the Debit Card TCL/IVR application and the audio files.

**call application voice** {*application_name*} {*language* | *redirect-number* | *pin-length* | *retry-count* | *set-location* | *uid-lenth* | *warning-time* }

### Example Syntax

**call application voice** <*.tcl script name*> **language 1 en**

OR

**call application voice** <*.tcl script name*> **set-location en 1** *tftp://keyer/debit audio/*

See also “” on page 33

## Syntax Description

| | |
|---|---|
| *application name* | The name of the application to which the parameters are being passed. |
| *param-name* | Name of the parameter that is being passed. |
| *param-value* | Value of the parameter that is being passed. |

The parameters used with this command are described in Table 6.

***Table 6 Valid Parameters***

| | |
|---|---|
| *language* | The number that identifies the language used by the audio files. For example, enter “1” for English language or “2” for Spanish language audio files. Any number can represent any languages.<br>Enter: **language** *digit language*.<br>For example: `call application voice test language 1 en`<br>Parameter values are:<br>• *number*— Tag that uniquely identifies an audio file. Enter 1 to indicate the primary language, and to 2 indicate the secondary language. Valid entries are from 0 - 9. (The digits 1 and 2 are hard coded to represent primary and secondary languages.)<br>• *language* — Two characters representing the language. For example, en for English, sp for Spanish and ch for Mandarin. Enter “aa” to represent all. |
| *redirect-number* | The designated operator telephone number of the service provider (or any other number designated by the customer). This is the number that calls are terminated to when debit time allowed has run out or the debit amount is exceeded. |
| *pin-length* | The number of characters in the PIN number. (0-10) |

*Table 6 Valid Parameters (continued)*

| | |
|---|---|
| *retry-count* | The number of times the caller is permitted to enter digits.<br>Enter **retry-count** *number* of retries. (1-5) |
| *set-location* | The location where the audio files are stored. This command connects both the *category* and *language* arguments that are used with the **call application voice** *language* command.<br>Enter: **set-location** *language* category *audio files URL or directory*<br>Parameter values are:<br>• location where audio files are stored. Enter the TFTP server location.<br>• *language* — en = English, ch = Mandarin, sp = Spanish.<br>• *category* — min = 0, max = 4 (0 means all).<br>• audio file URL or directory in the TFTP server.<br>The audio files can be divided into category groups (0 - 4). For example, audio files representing the days and months can be category 1, audio files representing units of currency can be category2, audio files representing units of time: seconds, minutes, and hours can be category 3.<br>For example: `set-location en 1 tftp://keyer/debitaudio/` |
| *uid-length* | The number of characters allowed to be entered for the user ID or PIN. (1-20) |
| *warning-time* | The number of seconds the user is warned before the allowed calling time has run out.<br>Enter **warning-time** *number* of seconds. (10-600) |

## Defaults

No default behavior or values. The system will not call an IVR application with the dial-peer.

Each variable must be defined for system interaction with the voice prompts.

## Command Modes

Dial-peer configuration mode.

## Command History

| Release | Modification |
|---|---|
| 12.0(7)T | This command was introduced. |
| 12.1(5)XM2 | The command was introduced for the Cisco AS5350 and CiscoAS5400. |

## Usage Guidelines

This command is used with the Debit Card feature.

Tips for use:

- **no call application voice** *app_name* removes the entire application and other parameters if configured.

- Adding **no** for any other line removes that particular parameter.
- **no call application voice** *app_name* **pin-len** will set the *pin-length* parameter to its default.
- The name of the application should be added to the dial-peer.

***Table 7 TCL Scripts and Parameters***

| TCL Script Name | Description - Summary | Parameters 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 command line interface (CLI). If the authentication fails, it allows the caller to retry. The retry number is also configured through the CLI. | • *uid-len* min = 1, max = 20, default - 10<br>• *pin-len* min = 0, max - 10, default = 4<br>• *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 command line interface (CLI). The account number and PIN are collected separately. | • *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 command line interface (CLI). The account number and PIN are collected separately. | • *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 command line interface (CLI). | • *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. | • *retry-count* min = 1, max = 5, default = 3 |

## Examples

Examples of syntax entered:

- The URL for (category 2, English language) audio files:

```
call application voice debitcard set-location en 2
tftp://keyer/faryaman/debitcard
```

- The URL for (all categories, all languages) audio files:

```
call application voice debitcard set-location aa 0
tftp://keyer/debitaudio/
```

Refer to Table 7 for the TCL script name and ensure that you have configured the corresponding parameters that are required for each TCL (those with CLI input) script to work.

- Example of sequence of commands entered during configuration:

```
Router(config) #call application voice prepaid tftp://keyer/debitcard.tcl


Router(config) #call application voice prepaid uid-len 4
Router(config) #call application voice prepaid language 1 en
Router(config) #call application voice prepaid language 2 sp
Router(config) #call application voice prepaid set-location en 0 tftp://keyer/

Router(config) #call application voice prepaid pin-len 4
Router(config) #call application voice prepaid retry-count 3
Router(config) #call application voice prepaid uid-len 10
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice** ***application_name location*** | Creates an application and indicates the location where the corresponding TCL files, which implement this application, are located. |
| **call application voice set-location** | Indicates where the audio files are stored. |

# call application voice *app-name* set-location

To indicate where the audio files are stored, use the **call application voice set-location** command. The file location needs to be the same as the location of the files that are linked to the category and language selection audio files used during configuration. This command is used by the Debit Card feature along with TCL scripts and audio files.

**call application voice** *app-name* **set-location** *language category location*

## Syntax Description

| | |
|---|---|
| *app-name* | The name of the application created to use for configuration. |
| **set-location** | The location where the ausio files are stored, usually a directory or URL site. This is a mandatory parameter. |
| *language* | A two-digit variable to specify the language used:<br>en = English, ch = Mandarin, sp = Spanish |
| *category* | A one-digit variable to specify the category.<br>The audio files can be divided into category groups (0 - 4). For example, audio files representing the days and months can be category 1, audio files representing units of currency can be category2, audio files representing units of time: seconds, minutes, and hours can be category 3.<br>For example: `set-location en 1 tftp://keyer/debitaudio/` |
| *location* | The TFTP server locationwhere audio files are stored. . |

## Defaults

No default behavior or values.

## Command Modes

Global configuration mode

## Command History

| Release | Modification |
|---|---|
| 12.0(7)T | This command was introduced. |
| 12.1(5)XM2 | The command was introduced for the Cisco AS5350 and CiscoAS5400. |

## Examples

```
call application voice debitcard set-location en 1 tftp://keyer/debitaudio/
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice** *application_name location* | Creates an application and indicates the location where the corresponding TCL files, which implement this application, are located. |
| **call application voice** *application_ name parameter-name parameter-value* | Passes parameters to the application created. |

# gw-accounting

To enable gateway specific accounting, use the **gw-accounting** command. There are three different methods of accounting. The H.323 method sends the CDR to the RADIUS server. The syslog method uses the system logging facility to record the CDRs. The VSA method collects VSA attributes. Use the **no** form of this command to disable gateway specific accounting.

**gw-accounting h323 [vsa] | syslog**

**no gw-accounting h323 [vsa] | syslog**

## Syntax Description

| | |
|---|---|
| **h323** | H.323 method uses RADIUS to output accounting CDRs. |
| **vsa** | (Optional) Vendor Specific Attributes are included in the RADIUS accounting. All the attributes that were included previously in the overloaded session ID are shown when this field is entered. See "VSA Attributes in Overloaded Session ID" in Table 1. |
| **syslog** | (Optional) Syslog uses the system logging facility to output CDRs. |

**Note** When using the Debit Card Cisco IOS feature you must use the H.323 VSA method of accounting.

## Defaults

Disable gateway specific accounting.

## Command Modes

Global configuration

## Command History

| Release | Modification |
|---|---|
| 11.3(6)NA2 | This command was introduced. |
| 12.0(7)T | The **vsa** field was added to this command. |
| 12.1(5)XM2 | The command was introduced for the Cisco AS5350 and CiscoAS5400. |

## Usage Guidelines

- This command is used when configuring the AAA accounting application.
- This command line defines a method for doing the accounting and enables the gateway to do the accounting. There are two accounting methods defined.
- Both **h.323** and **syslog** can be enabled at the same time, which causes CDRs to be generated in both methods.

## Related Commands

None.

# ivr autoload

To load files from TFTP to memory, use the **ivr autoload [mode verbose | silent] [retry** <*num*>] *url* <*index*> command. To disable this function, use the **no** form of this command.

**ivr autoload [mode verbose | silent] retry** *number* ***url*** *index*

**no ivr autoload [mode verbose | silent] retry** *number* ***url*** *index*

## Syntax Description

| | |
|---|---|
| **mode** | This is optional parameter. It specifies the mode. |
| **verbose** | Specifies to display the file transfer activity to the console. It is recommended for use while debugging. |
| **silent** | Performs the transfer in silent mode. No file transfer activity is displayed to the console. This is the default mode. |
| **retry** *number* | Specify the number of times to try transferring a file when there are errors. This parameter applies to each file transfer. This is an optional parameter.<br>*number* is an integer in the range of 1 - 5. Default is 3. |
| *url* | This parameter specifies the location of the index file. This is a mandatory parameter. The index file contains a list of all of the audio files. |
| *index* | This parameter specifies the location of the index file. This is a mandatory parameter. The index file contains a list of all of the audio files.<br>Example of index file on TFTP: `tftp://keyer/index`<br>Example of index file on Flash: `flash:index` |

## Defaults

See the defaults for individual parameters in Syntax Description.

## Command Modes

Global configuration

## Command History

| Release | Modification |
|---|---|
| 12.0(7)T | This command was introduced. |
| 12.1(5)XM2 | The command was introduced for the Cisco AS5350 and CiscoAS5400. |

## Usage Guidelines

An index file is provided as a parameter to this command. This file contains a list of audio files (URL) to be down loaded from TFTP. These audio files (prompts) that are loaded using **ivr autoload** command are not dynamically swapped out of memory. They are considered as autoloaded prompts as opposed to 'dynamic' prompts. (see the **ivr prompt memory** command for details on dynamic prompts.)

Issuing the **ivr autoload** command initiates the process of down-loading files from TFTP to memory. The command only starts up a back-ground process. The back-ground process (loader) does the actual down-loading of the files.

The back-ground process first reads the index file from either Flash or TFTP. It parses the files line by line looking for the URL. It ignores lines which starts with # as comment lines. Once it has a correct URL, it tries to read that .au file into memory and creates a media object. If there are any errors during the reading of the file, it retries the configured number of times. If *mode* is set to 'verbose' the loader logs the transaction to console. Once parsing has reached the end of the index file, the back-ground process exits out of memory.

Perform the following checks before initiating the back-ground process. If one of the checks fail, it indicates the back-ground process is not started, and instead you will see an error response to the command.

- Check if any prompt is being actively used (IVR is actively playing some prompts). If there are active prompts, the command fails with following error (.au files are also referred to as prompts).

  ```
  command is not allowed when prompts are active
  ```

- Check if there is already a back-ground process in progress. If there a process, command fails with following error.

  ```
  previous autoload command is still in progress
  ```

- Check if there is already a earlier **ivr autoload** command. If there is already an **ivr autoload** command configured,the user sees the following response when the command is issued.

  ```
  previous command is being replaced
  ```

- When the **no ivr autoload** command is issued, if there was already an **ivr autoload** command in progress, it will be aborted.

## Examples

```
ivr autoload mode verbose retry 3 url tftp://jurai/mgindi/tclware/index4
```

The index file for this example *index4* is shown as:

```
Router # more index4
tftp://jurai/mgindi/tclware/au/en/en_one.au
#tftp://jurai/mgindi/tclware/au/ch/ch_one.au
tftp://jurai/mgindi/tclware/au/ch/ch_one.au
```

## Related Commands

| Command | Description |
|---|---|
| **ivr prompt memory** | Configures the maximum amount of memory you wish to allow the dynamic audio files (prompts) to occupy in memory |

# ivr prompt memory

To configure the maximum amount of memory you wish to allow the dynamic audio files (prompts) to occupy in memory, use the **ivr prompt memory** *size* **files** *num* Global configuration command. Use the **no** form of this command to disable the maximum memory size.

**ivr prompt memory** *size* **files** *num*

**no ivr prompt memory** *size* **files** *num*

## Syntax Description

| | |
|---|---|
| *size* | Specify an integer which specifies the maximum memory to be used by the free dynamic prompts. Unit is Kilobytes.<br>Min. = 128<br>Max. = 16 MB<br>Default = 3MB |
| **files** | This is an optional parameter. |
| *num* | Enter an integer which specifies the maximum number of free dynamic prompts that would stay in memory.<br>Min. = 1<br>Max. = 1000<br>Default = 200 |

## Defaults

See individual parameter defaults in Syntax Description

## Command Modes

Global configuration

## Command History

| Release | Modification |
|---|---|
| 12.0(7)T | This command was introduced. |
| 12.1(5)XM2 | The command was introduced for the Cisco AS5350 and CiscoAS5400. |

## Usage Guidelines

When both the *number* and *size* parameters are specified, the minimum memory out of the two will be used for memory calculations.

All the prompts which are not autoloaded or fixed are considered as dynamic. Dynamic prompts are loaded in to memory from TFTP or Flash, as and when they are needed. When they are actively used for playing prompts they are considered to be in 'active' state. However, once the prompt playing is complete, these prompts are no more active and are considered to be in 'free' state.

The free prompts either stay in memory or removed out of memory depending on the availability of space in memory for these free prompts. The **prompt-mem** command essentially specifies a maximum memory to be used for these free prompts.

The free prompts are saved in the memory and are queued in a waitQ. When the waitQ is full (either because the totally memory occupied by the free prompts exceeds the max. configured value or the number of files in the waitQ exceeds max. configured), oldest free prompts are removed out of memory.

## Examples

```
ivr prompt memory 2048 files 500
```

## Related Commands

| Command | Description |
|---|---|
| **ivr autoload** | Loads the files from TFTP to memory. |
| **show call prompt-mem-usage** | Displays the memory size use by prompts. |

# Glossary

**AAA**—Authentication, Authorization, and Accounting.

**CLI**—Command Line Interface

**DTMF**—Dual tone multi frequency. Use of two simultaneous voice-band tones for dialing (such as touch tone).

**IVR**—Interactive Voice Response. The system plays a message to the calling party and may also collect information from the calling party in order to interact with the caller.

**NTP**—Network Time Protocol. NTP is a protocol built on top of TCP that assures accurate local time-keeping with reference to radio and atomic clocks located on the Internet. This protocol is capable of synchronizing distributed clocks within milliseconds over long time periods.

**On-net**—On-net calls are voice over IP calls processed on the IP network.

**Off-net**—Off-net calls are those that are circuit-switched calls, whether switched by the PSTN or a PBX.

**POP**—Point of Presence. The location where a call enters and exits the VoIP interexchange network from and to the PSTN.

**PSTN**—Public Switched Telephone Network.

**TCL**—Tool Command Language. TCL is an interpreted script language developed by Dr. John Ousterhout at the University of California, Berkeley, and now developed and maintained by Sun Microsystems Laboratories.

**VoIP**—Voice over IP

**VSA**—Vendor Specific Attributes.