

# package through pattern


- package, on page 3
- package appcommon, on page 5
- package callsetup, on page 6
- package language, on page 7
- package persistent, on page 9
- package session_xwork, on page 11
- param, on page 12
- param access-method, on page 15
- param account-id-method, on page 16
- param accounting enable, on page 18
- param accounting-list, on page 19
- param authen-list, on page 21
- param authen-method, on page 22
- param authentication enable, on page 24
- param convert-discpi-after-connect, on page 25
- param dsn-script, on page 27
- param event-log, on page 28
- param fax-dtmf, on page 30
- param global-password, on page 31
- param language, on page 32
- param mail-script, on page 34
- param mode, on page 36
- param pin-len, on page 38
- param prompt, on page 40
- param redirect-number, on page 42
- param reroutemode, on page 44
- param retry-count, on page 46
- param security, on page 48
- param uid-len, on page 50
- param voice-dtmf, on page 52
- param warning-time, on page 53
- paramspace, on page 55
- paramspace appcommon event-log, on page 57

- paramspace appcommon security, on page 59
- paramspace callsetup mode, on page 61
- paramspace callsetup reroutemode, on page 63
- paramspace language, on page 65
- paramspace session_xwork convert-discpi-after-connect, on page 67
- pass-thru content, on page 69
- pass-thru headers, on page 71
- passthru-hdr, on page 72
- passthru-hdr-unsupp, on page 74
- pattern, on page 75

# package

To enter application-parameter configuration mode to load and configure a package, use the **package** command in application configuration mode. There is no **no** form of this command.

**package** *package-name location*
**no package** *package-name*

## Syntax Description

| | |
|---|---|
| *package-name* | Name that identifies the package. |
| location | Directory and filename of the package in URL format. For example, flash memory (flash:filename), a TFTP (tftp://../filename) or an HTTP server (http://../filename) are valid locations. |

## Command Default

No default behavior or values

## Command Modes

Application configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced. |

## Usage Guidelines

Use this command to enter application parameter configuration mode to load and configure a package. A package is a linkable set of C or Tcl functions that provide functionality invoked by applications or other packages. They are not standalone. For example, a debit card application may use multiple language translation packages, such as English and French. These language translation packages can also be used by other applications without having to modify the package for each application using it.

The packages available on your system depend on the scripts, applications, and packages that you have installed. Your software comes with a set of built-in packages, and additional packages can be loaded using the Tcl **package** command. You can then use the **package** command in application configuration mode to access the parameters contained in those packages.

## Examples

The following example shows that a French language translation package is loaded:

```
Router(config-app)# package frlang http://server-1/language_translate.tcl
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice** | Defines the name of a voice application and specify the location of the Tcl or VoiceXML document to load for this application. |
| **package appcommon** | Configures parameters in the built-in common voice application package. |
| **package callsetup** | Configures parameters in the built-in call setup package. |
| **package language** | Loads an external Tcl language module for use with an IVR application. |

| Command | Description |
| --- | --- |
| **package session_xwork** | Configure parameters in the built-in session_xwork package. |

# package appcommon

To configure parameters in the built-in common voice application package, use the **package appcommon** command in application configuration mode. There is no **no** form of this command.

**package appcommon**

## Syntax Description

No arguments or keywords

## Command Default

No default behavior or values

## Command Modes

Application configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced. |

## Usage Guidelines

Use this command to configure common voice-application-package parameters. After you enter this command, use the **param** command to configure individual parameters.

## Examples

The following example shows using the **param security trusted** command to set the security level of a VoiceXML application to "trusted" so that automatic number identification (ANI) is not blocked.

```
application
package appcommon
param security trusted
```

## Related Commands

| Command | Description |
|---|---|
| **package** | Enters application parameter configuration mode to load and configure a package. |
| **package callsetup** | Configures parameters in the built-in call setup package. |
| **package language** | Loads an external Tcl language module for use with an IVR application. |
| **package session_xwork** | Configures parameters in the built-in session_xwork package. |

# package callsetup

To configure parameters in the built-in call setup package, use the **package callsetup**command in application configuration mode. There is no **no** form of this command.

**package callsetup**

**Command Default** No arguments or keywords

**Command Default** No default behavior or values

**Command Modes**

Application configuration

**Command History**

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced. |

**Usage Guidelines** Use this command to configure parameters in the built-in call setup package. The callsetup package is used by applications and other packages to place outbound call legs and interwork them with incoming call legs. call setup After you enter this command, use the **param** command to configure individual parameters.

**Examples**

```
The following example shows the call transfer mode set to redirect:
application
package callsetup
param mode redirect
```

**Related Commands**

| Command | Description |
|---|---|
| **package** | Enters application parameter configuration mode to load and configure a package. |
| **package appcommon** | Configures parameters in the built-in common voice application package. |
| **package language** | Loads an external Tcl language module for use with an IVR application. |
| **package session_xwork** | Configure parameters in the built-in session_xwork package. |

# package language

To load an external Tool Command Language (Tcl) language module for use with an interactive voice response (IVR) application, use the **package language** command in application configuration mode.

**package language** *prefix url*

## Syntax Description

| | |
|---|---|
| *prefix* | Two-character prefix for the language; for example, "*en"* for English or "*ru"* for Russian. |
| *url* | Location of the module. |

## Command Default

No default behavior or values

## Command Modes

Application configuration (config-app)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE 26.2.1 | This command was modified to display warning message for insecure protocol usage. |
| 12.3(14)T | This command was introduced to replace the **call language voice** command. |

## Usage Guidelines

Use this command to load language packages for use by applications or other packages. The built-in languages are English (*en)* , Chinese (*ch)* , and Spanish (*sp)* . If you specify "*en"* , "*ch"* , or "*sp"* , the new Tcl module replaces the built-in language functionality. When you add a new Tcl module, you create your own prefix to identify the language. When you configure and load the new languages, any upper-layer application (Tcl IVR) can use the language.

After loading language packages, you can configure an application or other package to use the new language package using the **param language** or **paramspace language location** command.

**Warning** Starting with Cisco IOS XE 26.2.1, a warning message is displayed when insecure protocols such as HTTP, FTP, or TFTP are used, as they do not provide encryption or authentication. Use secure alternatives such as HTTPS, SFTP, or SCP instead.

## Examples

The following example adds Russian (*ru* ) as a Tcl module and configures the debitcard application to use Russian for prompts:

```
application
package language ru tftp://box/unix/scripts/multi-lang/ru_translate.tcl
service debitcard tftp://server-1/tftpboot/scripts/app_debitcard.2.0.2.8.tcl
param language ru
```

The following example shows the warning message displayed for the usage of insecure HTTP protocol:

```
Device(config)#application
Device(config-app)#package language http://10.10.10.70//tftpboot/COLT-DE-CUCM.cfg
SECURITY WARNING - Module: CONFIG_VOICE, Command: package language
http://10.10.10.70//tftpboot/COLT-DE-CUCM.cfg,
Reason: Legacy protocol poses data confidentiality and integrity risks due to lack of
encryption and authentication,
Description: package language configured with FTP/HTTP/TFTP URL protocol,
Remediation: Use secure protocols such as HTTPS, SFTP, or SCP instead of insecure HTTP,
FTP, or TFTP
Device(config-app-param)#
```

## Related Commands

| Command | Description |
|---|---|
| **package** | Enters application parameter configuration mode to load and configure a package. |
| **package appcommon** | Configures parameters in the built-in common voice application package. |
| **package callsetup** | Configures parameters in the built-in call setup package. |
| **package session_xwork** | Configures parameters in the built-in session_xwork package. |
| **param language** | Configures the language parameter in a service or package on the gateway. |
| **paramspace language location** | Defines the category and location of audio files that are used for dynamic prompts by an IVR application (Tcl or VoiceXML). |

# package persistent

To configure the package type used when reporting persistent events for a multifrequency (MF) tone channel-associated signaling (CAS) endpoint type using a specific Media Gateway Control Protocol (MGCP) profile, use the **package persistent** command in MGCP profile configuration mode. To disable the persistent status, use the **no** form of this command.

**package persistent** *package-name*
**no package persistent** *package-name*

## Syntax Description

| | |
|---|---|
| *package -name* | Package name. Valid names are ms-package and mt-package. |

## Command Default

ms-package

## Command Modes

MGCP profile configuration

## Command History

| Release | Modification |
|---|---|
| 12.2(2)XA | This command was introduced. |
| 12.2(4)T | This command was integrated into Cisco IOS Release 12.2(4)T. |
| 12.2(11)T | This command was implemented on the Cisco AS5300 and Cisco AS5850. |

## Usage Guidelines

This command is used when configuring values for a MGCP profile.

This command is used only with MF trunks (gateway voice ports configured with the **dial-type mf** command in voice-port configuration mode). Because the same persistent event can be defined in different MGCP packages, you may need to use this command to tell the gateway which package to use when reporting persistent events to the call agent for the endpoints in this MGCP profile. For example, a T1 may be configured as an MF trunk, but there is more than one MGCP package that applies to an MF trunk. An *ans* (call answer) event must be mapped to the appropriate package for call-agent notification. This command allows different T1s to be configured for different CAS protocols.

The MS package is used with certain PBX direct inward dial (DID) and direct outward dial (DOD) trunks with wink-start or ground-start signaling as indicated in RFC 3064 (*MGCP CAS Packages* ).

The MT package is a subset of the MS package, and it is used with certain operator services on terminating MF trunks on trunking gateway endpoints, as described in *PacketCable PSTN Gateway Call Signaling Protocol Specification* (TGCP) PKT-SP-TGCP-D02-991028, December 1, 1999.

## Examples

The following example enables event persistence for the MT package:

```
Router(config)# mgcp profile nyc-ca
Router(config-mgcp-profile)# package persistent mt-package
```

**Related Commands**

| Command | Description |
|---|---|
| **mgcp** | Starts and allocates resources for the MGCP daemon. |
| **mgcp profile** | Initiates MGCP profile mode to create and configure an MGCP profile associated with one or more endpoints or to configure the default profile. |

# package session_xwork

To configure parameters in the built-in session_xwork package, use the **package session_xwork**command in application configuration mode.

**package session_xwork**

## Syntax Description

No arguments or keywords

## Command Default

No default behavior or values

## Command Default

Application configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced. |

## Usage Guidelines

Use this command to configure parameters in the built-in session x_work package. After you enter this command, use the **param** command to configure individual parameters.

For example, use this command with the **param default disc-prog-ind-at-connect** command to convert a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state.

## Examples

The following example shows how to configure the system to convert a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state:

```
application
package session_xwork
param default disc-prog-ind-at-connect
```

## Related Commands

| Command | Description |
|---|---|
| **package** | Enters application parameter configuration mode to load and configure a package. |
| **package appcommon** | Configures parameters in the built-in common voice application package. |
| **package callsetup** | Configures parameters in the built-in call setup package. |
| **package language** | Loads an external Tool Command Language (Tcl) language module for use with an interactive voice response (IVR) application. |
| **param convert-discpi-after-connect** | Enables or disables conversion of a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |

# param

To load and configure parameters in a package or a service (application) on the gateway, use the **param** command in application configuration mode. To reset a parameter to its default value, use the **no** form of this command.

**param** *param-name* [**param max-retries** | **param passwd** | **param passwd-prompt filename** | **param user-prompt filename** | **param term-digit** | **param abort-digit** | **param max-digits**]
**no param** *param-name*

## Syntax Description

| | |
|---|---|
| *param-name* | Name of the parameter. |
| param max-retries | (Optional) Number of attempts to re-enter account or password. Value ranges from 0-10, default value is 0. |
| param passwd | (Optional) Character string that defines a predefined password for authorization. |
| param passwd-prompt filename | (Optional) Announcement URL to request password input. filename defines the name and location of the audio filename to be used for playing the password prompt. |
| param user-prompt filename | (Optional) Announcement URL to request authorization code username. filename defines the name and location of the audio filename to be used for playing the username prompt. |
| param term-digit | Digit for terminating username or password digit input. |
| param abort-digit | Digit for aborting username or password digit input. Default value is *. |
| param max-digits | Maximum number of digits in a username or password. Range of valid value: 1 - 32. Default value is 32. |

## Command Default

No default behavior or value.

## Command Modes

Application configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced. |
| 15.1(3)T | This command was modified. The following keywords and arguments were added: param max-retries, param passwd, param passwd-prompt filename, param user-prompt filename, param term-digit, param max-digit. |

## Usage Guidelines

Use this command in application parameter configuration mode to configure parameters in a package or service. A package is a linkable set of C or Tcl functions that provide functionality invoked by applications or other packages. A service is a standalone application.

The parameters available for configuration differ depending on the package or service that is loaded on the gateway. The **param register** Tcl command in a service or package registers a parameter and provides a description and default values which allow the parameter to be configured using the CLI. The **param register** command is executed when the service or package is loaded or defined, along with commands such as **package provide**, which register the capability of the configured module and its associated scripts. You must configure and load the Tcl scripts for your service or package and load the package in order to configure its parameters. See the *Tcl IVR API Version 2.0 Programming Guide* for more information.

When a package or service is defined on the gateway, the parameters in that package or service become available for configuration when you use this command. Additional arguments and keywords are available for different parameters. To see a list of available parameters, enter **param** ?.

To avoid problems with applications or packages using the same parameter names, the *parameter namespace,* or *parameterspace* concept is introduced. When a service or a package is defined on the gateway, its parameter namespace is automatically defined. This is known as the service or package's local parameterspace, or "myparameterspace." When you use this command to configure a service or package's parameters, the parameters available for configuration are those contained in the local parameterspace. If you want to use parameter definitions found in different parameterspace, you can use the **paramspace***parameter-namespace* command to map the package's parameters to a different parameterspace. This allows that package to use the parameter definitions found in the new parameterspace, in addition to its local parameterspace.

Use this command in Cisco Unified Communication Manager Express 8.5 and later versions to define the username and password parameters to authenticate packages for Forced Authorization Code (FAC)

When a predefined password is entered using the param passwd keyword, callers are not requested to enter a password. You must define a filename for user-prompt to play an audio prompt requesting the caller to enter a valid username (in digits) for authorization. Similarly, you must define a filename for passwd-prompt to play an audio prompt requesting the caller to enter a valid password (in digits) for authorization.

## Examples

The following example shows how to configure a parameter in the httpios package:

```
application
package httpios
param paramA value4
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice** | Defines the name of a voice application and specify the location of the Tcl or VoiceXML document to load for this application. |
| **param account-id-method** | Configures an application to use a particular method to assign the account identifier. |
| **param convert-discpi-after-connect** | Enables or disables conversion of a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |
| **param event-log** | Enables or disables logging for linkable Tcl functions (packages). |
| **param language** | Configures the language parameter in a service or package on the gateway. |
| **param mode** | Configures the call transfer mode for a package. |

| Command | Description |
|---|---|
| **param pin-len** | Defines the number of characters in the personal identification number (PIN) for an application. |
| **param redirect-number** | Defines the telephone number to which a call is redirected--for example, the operator telephone number of the service provider--for an application. |
| **param reroutemode** | Configures the call transfer reroutemode (call forwarding) for a package. |
| **param retry-count** | Defines the number of times a caller is permitted to reenter the PIN for a designated application and passes that information to the application. |
| **param security** | Configures security for linkable Tcl functions (packages). |
| **paramspace** | Enables an application to use parameters from the local parameter space of another application. |
| **param uid-length** | Defines the number of characters in the UID for a package. |
| **param warning-time** | Defines the number of seconds of warning that a user receives before the allowed calling time expires. |

# param access-method

To specify the access method for two-stage dialing for the designated application, use the **param access-method**command in application parameter configuration mode. To restore default values for this command, use the **no** form of this command.

**param access-method** {**prompt-user** | **redialer**}
**no param access-method**

## Syntax Description

| | |
|---|---|
| **prompt-user** | Specifies that no DID is set in the incoming POTS dial peer and that a Tcl script in the incoming POTS dial peer is used for two-stage dialing. |
| **redialer** | Specifies that no DID is set in the incoming POTS dial peer and that the redialer device are used for two-stage dialing. |

## Command Default

Prompt-user (when DID is not set in the dial peer)

## Command Modes

Application parameter configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice access-method** command. |

## Usage Guidelines

Use the **param access-method** command to specify the access method for two-stage dialing when DID is disabled in the POTS dial peer.

## Examples

The following example specifies prompt-user as the access method for two-stage dialing for the app_libretto_onramp9 IVR application:

```
application
service app_libretto_onramp9 tftp://server-1/tftpboot/scripts
param access-method prompt-user
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice access-method** | Specifies the access method for two-stage dialing for the designated application. |

# param account-id-method

To configure an application to use a particular method to assign the account identifier, use the **param account id method**command in application parameter configuration mode. To remove configuration of this account identifier, use the **no** form of this command.

**param account-id-method** {**none** | **ani** | **dnis** | **gateway**}
**no param account-id-method** {**none** | **ani** | **dnis** | **gateway**}

## Syntax Description

| | |
|---|---|
| **none** | Account identifier is blank. This is the default. |
| **ani** | Account identifier is the calling party telephone number (automatic number identification, or ANI). |
| **dnis** | Account identifier is the dialed party telephone number (dialed number identification service, or DNIS). |
| **gateway** | Account identifier is a router-specific name derived from the hostname and domain name, displayed in the following format: router-name.domain-name. |

## Command Default

No default behavior or values

## Command Modes

Application parameter configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice account-id-method** command. |

## Usage Guidelines

When an on-ramp application converts a fax into an e-mail, the e-mail contains a field called x-account-id, which can be used for accounting or authentication. The x-account-id field can contain information supplied as a result of this command, such as the calling party's telephone number (**ani**), the called party's telephone number (**dnis**), or the name of the gateway (**gateway**).

## Examples

The following example sets the fax detection IVR application account identifier to the router-specific name derived from the hostname and domain name:

```
application
service fax_detect flash:app_fax_detect.2.1.2.2.tcl
param account-id-method gateway
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice account-id-method** | Configures the fax detection IVR application to use a particular method to assign the account identifier. |
| **param** | Loads and configures parameters in a package or a service (application). |

| Command | Description |
|---|---|
| **param convert-discpi-after-connect** | Enables or disables conversion of a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |
| **param event-log** | Enables or disables logging for linkable Tcl functions (packages). |
| **param language** | Configures the language parameter in a service or package on the gateway. |
| **param mode** | Configures the call transfer mode for a package. |
| **param pin-len** | Defines the number of characters in the PIN for an application. |
| **param redirect-number** | Defines the telephone number to which a call is redirected--for example, the operator telephone number of the service provider--for an application. |
| **param reroutemode** | Configures the call transfer reroutemode (call forwarding) for a package. |
| **param retry-count** | Defines the number of times a caller is permitted to reenter the PIN for a designated application and passes that information to the application. |
| **param security** | Configures security for linkable Tcl functions (packages). |
| **param uid-length** | Defines the number of characters in the UID for a package. |
| **param warning-time** | Defines the number of seconds of warning that a user receives before the allowed calling time expires. |

# param accounting enable

To enable authentication, authorization, and accounting (AAA) accounting for a Tool Command Language (TCL) application, use the **param accounting enable**command in application configuration mode. To disable accounting for a TCL application, use the **no** form of this command.

**param accounting enable**
**no param accounting enable**

## Syntax Description

This command has no arguments or keywords.

## Command Default

Disabled

## Command Modes

Application configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice accounting enable**command. |

## Usage Guidelines

This command enables AAA accounting services if a AAA accounting method list has been defined using both the **aaa accounting** command and the **mmoip aaa method fax accounting** command.

This command applies to off-ramp store-and-forward fax functions.

## Examples

The following example enables AAA accounting to be used with outbound store-and-forward fax:

```
application
service app_libretto_onramp9 tftp://server-1/tftpboot/scripts/
param accounting enable
```

## Related Commands

| Command | Description |
|---|---|
| **aaa accounting** | Enables AAA accounting of requested services when you use RADIUS or TACACS+. |
| **mmoip aaa method fax accounting** | Defines the name of the method list to be used for AAA accounting with store-and-forward fax. |

# param accounting-list

To define the name of the accounting method list to be used for authentication, authorization, and accounting (AAA) with store-and-forward fax on a voice feature card (VFC), use the **param accounting list** command in application configuration mode. To undefine the accounting method list, use the **no** form of this command.

**param accounting-list** *method-list-name*
**no param accounting-list** *method-list-name*

## Syntax Description

| | |
|---|---|
| *method-list-name* | Character string used to name a list of accounting methods to be used with store-and-forward fax. |

## Command Default

No AAA accounting method list is defined

## Command Modes

Application configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | The **param accounting-list command** was introduced to replace the **call application voice accounting-list**command. |

## Usage Guidelines

This command defines the name of the AAA accounting method list to be used with store-and-forward fax. The method list itself, which defines the type of accounting services provided for store-and-forward fax, is defined using the **aaa accounting** command. Unlike standard AAA (in which each defined method list can be applied to specific interfaces and lines), the AAA accounting method lists that are used in store-and-forward fax are applied globally.

After the accounting method lists have been defined, they are enabled by using the **mmoip aaa receive accounting enable**command.

This command applies to both on-ramp and off-ramp store-and-forward fax functions on VFCs. The command is not used on modem cards.

## Examples

The following example defines a AAA accounting method list "smith" to be used with store-and-forward fax:

```
aaa new-model
application
service app_libretto_onramp9 tftp://server-1/tftpboot/scripts/
param accounting-list smith
```

## Related Commands

| Command | Description |
|---|---|
| **aaa accounting** | Enables AAA accounting of requested services when you use RADIUS or TACACS+. |
| **param accounting enable** | Enables AAA accounting for a TCL application. |

| Command | Description |
| --- | --- |
| **mmoip aaa receive-accounting enable** | Enables on-ramp AAA accounting services. |

# param authen-list

To specify the name of an authentication method list for a Tool Command Language (TCL) application, use the **param authen list**command in global configuration mode. To disable the authentication method list for a TCL application, use the **no** form of this command.

**param authen-list** *method-list-name*
**no param authen-list** *method-list-name*

## Syntax Description

| | |
|---|---|
| *method-list-name* | Character string used to name a list of authentication methods to be used with T.38 fax relay and T.37 store-and-forward fax. |

## Command Default

No default behavior or values

## Command Modes

Application configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice param authen-list** command. |

## Usage Guidelines

This command defines the name of the authentication, authorization, and accounting (AAA) method list to be used with fax applications on voice feature cards. The method list itself, which defines the type of authentication services provided for store-and-forward fax, is defined using the **aaa authentication** command. Unlike standard AAA (in which each defined method list can be applied to specific interfaces and lines), AAA method lists that are used with fax applications are applied globally.

After the authentication method lists have been defined, they are enabled by using the **param authentication enable**command.

## Examples

The following example defines a AAA authentication method list (called "fax") to be used with T.38 fax relay and T.37 store-and-forward fax:

```
application
service app_libretto_onramp9 tftp://server-1/tftpboot/scripts/
param authen-list fax
param authentication enable
```

## Related Commands

| Command | Description |
|---|---|
| **aaa authentication** | Enable AAA accounting of requested services for billing or security purposes. |
| **param authen-method** | Specifies the authentication method for a TCL application. |
| **param authentication enable** | Enables AAA authentication services for a TCL application. |

# param authen-method

To specify an authentication, authorization, and accounting (AAA) authentication method for a Tool Command Language (Tcl) application, use the **param authen-method**command in application configuration mode. To disable the authentication method for a Tcl application, use the **no** form of this command.

**param authen-method** {**prompt-user** | **ani** | **dnis** | **gateway** | **redialer-id** | **redialer-dnis**}
**no param authen-method** {**prompt-user** | **ani** | **dnis** | **gateway** | **redialer-id** | **redialer-dnis**}

## Syntax Description

| | |
|---|---|
| **prompt user** | User is prompted for the Tcl application account identifier. |
| **ani** | Calling party telephone number (automatic number identification or ANI) is used as the Tcl application account identifier. |
| **dnis** | Called party telephone number (dialed number identification service or DNIS) is used as the Tcl application account identifier. |
| **gateway** | Router-specific name derived from the host name and domain name is used as the Tcl application account identifier, displayed in the following format: *router-name.domain-name*. |
| **redialer id** | Account string returned by the external redialer device is used as the Tcl application account identifier. In this case, the redialer ID is either the redialer serial number or the redialer account number. |
| **redialer dnis** | Called party telephone number (dialed number identification service or DNIS) is used as the Tcl application account identifier captured by the redialer if a redialer device is present. |

## Command Default

No default behavior or values

## Command Modes

Application configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice authen-method** command in application configuration mode. |

## Usage Guidelines

Normally, when AAA is used for simple user authentication, AAA uses the username information defined in the user profile for authentication. With T.37 store-and-forward fax and T.38 real-time fax, you can specify that the ANI, DNIS, gateway ID, redialer ID, or redialer DNIS be used to identify the user for authentication or that the user be prompted for the Tcl application.

## Examples

The following example configures the router-specific name derived from the host name and domain name as the Tcl application account identifier for the app_libretto_onramp9 Tcl application:

```
application
service app_libretto_onramp9 tftp://server-1/tftpboot/scripts/
param authen-method gateway
```

**Related Commands**

| Command | Description |
|---|---|
| **param authentication enable** | Enables AAA authentication services for a Tcl application. |

# param authentication enable

To enable authentication, authorization, and accounting (AAA) services for a Tool Command Language (TCL) application, use the **param authentication enable**command in application configuration mode. To disable authentication for a TCL application, use the **no** form of this command.

**param authentication enable**
**no param authentication enable**

## Syntax Description

This command has no arguments or keywords.

## Command Default

No default behavior or values

## Command Modes

Application configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice authentication enable**command. |

## Usage Guidelines

This command enables AAA authentication services for a TCL application if a AAA authentication method list has been defined using the **aaa authentication** command and the **param authen-list** command.

## Examples

The following example enables AAA authentication for an authentication method list (called "fax") with outbound store-and-forward fax.

```
application
service app_libretto_onramp9 tftp://server-1/tftpboot/scripts/
param authen-list fax
param authentication enable
```

## Related Commands

| Command | Description |
|---|---|
| **aaa authentication** | Enables AAA accounting of requested services when you use RADIUS or TACACS+. |
| **param authen-list** | Specifies the name of an authentication method list for a Tool Command Language (TCL) application. |
| **param authen-method** | Specifies the authentication method for a TCL application. |

# param convert-discpi-after-connect

To enable or disable conversion of a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state, use the **param convert-discpi-after-connect**command in application parameter configuration mode. To restore this parameter to the default value, use the **no** form of this command.

**param convert-discpi-after-connect** {**enable** | **disable**}
**no param convert-discpi-after-connect** {**enable** | **disable**}

## Syntax Description

| | |
|---|---|
| **enable** | Convert a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |
| **disable** | Revert to a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) when the call is in the active state. |

## Command Default

Enabled

## Command Modes

Application parameter configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice default disc-prog-ind-at-connect** command. |

## Usage Guidelines

This command has no effect if the call is not in the active state. This command is available for the session_xwork package. If you are configuring this parameter for a package, you must first use the command **package session x_work**.

If you are configuring this parameter for a service, use the following commands:

**service** *name url*

**paramspace session_xwork convert-discpi-after-connect**

## Examples

The following example shows conversion enabled for a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8):

```
application
package session_xwork
param convert-discpi-after-connect enable
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice default disc-prog-ind-at-connect** | Converts a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |

| Command | Description |
|---|---|
| **param** | Loads and configures parameters in a package or a service (application). |
| **param account-id-method** | Configures an application to use a particular method to assign the account identifier. |
| **param event-log** | Enables or disables logging for linkable Tcl functions (packages). |
| **param language** | Configures the language parameter in a service or package on the gateway. |
| **param mode** | Configures the call transfer mode for a package. |
| **param pin-len** | Defines the number of characters in the personal identification number (PIN) for an application. |
| **param redirect-number** | Defines the telephone number to which a call is redirected--for example, the operator telephone number of the service provider--for an application. |
| **param reroutemode** | Configures the call transfer reroutemode (call forwarding) for a package. |
| **param retry-count** | Defines the number of times a caller is permitted to reenter the PIN for a designated application and passes that information to the application. |
| **param security** | Configures security for linkable Tcl functions (packages). |
| **param uid-length** | Defines the number of characters in the UID for a package. |
| **param warning-time** | Defines the number of seconds of warning that a user receives before the allowed calling time expires. |

# param dsn-script

To specify the VoiceXML application to which the off-ramp mail application hands off calls for off-ramp delivery status notification (DSN) and message disposition notification (MDN) e-mail messages, use the **param dsn script**command in application parameter configuration mode. To remove the application, use the **no** form of this command.

**param dsn-script** *application-name*
**no param dsn-script** *application-name*

## Syntax Description

| | |
|---|---|
| *application-name* | Name of the VoiceXML application to which the off-ramp mail application hands off the call when the destination answers. |

## Command Default

No default behavior or values

## Command Modes

Application parameter configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice dsn-script** command. |

## Usage Guidelines

When the off-ramp gateway receives a DSN or MDN e-mail message, it handles it in the same way as a voice e-mail trigger message. The dial peer is selected on the basis of dialed number identification service (DNIS), and the mail application hands off the call to the VoiceXML application that is configured with this command.

## Examples

The following example shows how to define the DSN application and how to apply it to a dial peer:

```
application
service offramp-mapp tftp://sample/tftp-users/tcl/app_voicemail_offramp.tcl
param dsn-script dsn-mapp-test
!
dial-peer voice 1000 mmoip
 application offramp-mapp
 incoming called-number 555....
 information-type voice
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice dsn-script** | Specifies the VoiceXML application to which the off-ramp mail application hands off calls for off-ramp DSN and MDN e-mail messages. |

# param event-log

To enable or disable logging for linkable Tcl functions (packages), use the **param event-log** command in application parameter configuration mode. To restore this parameter to the default value, use the **no** form of this command.

**param event-log** {**enable** | **disable**}
**no param event-log** {**enable** | **disable**}

## Syntax Description

| | |
|---|---|
| **enable** | Event logging is enabled. |
| **disable** | Event logging is disabled. |

## Command Default

No default behavior or values

## Command Modes

Application parameter configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice event-log** command. |

## Usage Guidelines

This command is available for the built-in common voice application package. If you are configuring this parameter for that package, you must first use the command **package appcommon**.

If you are configuring this parameter for a service, use the following commands:

**service** *name url*

**paramspace appcommon event-log**

If you are configuring event logging for all voice applications, use the **event-log** command in application configuration monitor mode.

**Note** To prevent event logging from adversely impacting system resources for production traffic, the gateway uses a throttling mechanism. When free processor memory drops below 20%, the gateway automatically disables all event logging. It resumes event logging when free memory rises above 30%. While throttling is occurring, the gateway does not capture any new event logs even if event logging is enabled. You should monitor free memory and enable event logging only when necessary for isolating faults.

## Examples

The following example shows event-logging disabled for the built-in common voice application package:

```
application
package appcommon
param event-log disable
```

**Related Commands**

| Command | Description |
|---|---|
| **call application voice event-log** | Enables event logging for a specific voice application. |
| **event-log** | Enables event logging for applications. |
| **package appcommon** | Configures parameters in the built-in common voice application package. |
| **param** | Loads and configures parameters in a package or a service (application). |
| **param account-id-method** | Configures an application to use a particular method to assign the account identifier. |
| **param convert-discpi-after-connect** | Enables or disables conversion of a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |
| **param language** | Configures the language parameter in a service or package on the gateway. |
| **param mode** | Configures the call transfer mode for a package. |
| **param pin-len** | Defines the number of characters in the PIN for an application. |
| **param redirect-number** | Defines the telephone number to which a call is redirected--for example, the operator telephone number of the service provider--for an application. |
| **param reroutemode** | Configures the call transfer reroutemode (call forwarding) for a package. |
| **param retry-count** | Defines the number of times a caller is permitted to reenter the PIN for a designated application and passes that information to the application. |
| **param security** | Configures security for linkable Tcl functions (packages). |
| **param uid-length** | Defines the number of characters in the UID for a package. |
| **param warning-time** | Defines the number of seconds of warning that a user receives before the allowed calling time expires. |

# param fax-dtmf

To direct the fax detection interactive voice response (IVR) application to recognize a specified digit to indicate a fax call in default-voice and default-fax modes, use the **param fax-dtmf** command in application parameter configuration mode. To remove configuration of this digit, use the **no** form of this command.

**param fax-dtmf** {**0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | ***** | **#**}
**no param fax-dtmf** {**0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | ***** | **#**}

## Syntax Description

| | |
|---|---|
| **0** \| **1**\| **2**\| **3**\| **4**\| **5**\| **6**\| **7**\| **8**\| **9**\| *****\| **#** | The telephone keypad digit processed by the calling party to indicate a fax call, in response to the audio prompt that plays during the default-voice or default-fax mode of the fax detection IVR application. |

## Command Default

**2**

## Command Modes

Application parameter configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command is introduced to replace the **call application voice fax-dtmf** command. |

## Usage Guidelines

This command is useful only when the fax detection IVR application is being configured in default-voice mode or default-fax mode as defined by the **param mode** command.

If you also configure voice DTMF using the **param voice-dtmf** command, you must use different numbers for the voice and fax DTMF digits.

## Examples

The following example selects DTMF digit 1 to indicate a fax call:

```
application
service faxdetect tftp://sample/tftp-users/tcl/app_fax_detect.2.x.x.tcl
param fax-dtmf 1
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice fax-dtmf** | Directs the fax detection IVR application to recognize a specified digit to indicate a fax call in default-voice and default-fax modes. |
| **param mode** | Configures the call transfer mode for a package. |
| **param voice-dtmf** | Directs an application to recognize a specified digit to indicate a voice call in default-voice and default-fax modes. |

# param global-password

To define a password to be used with CiscoSecure for Windows NT when using store-and-forward fax on a voice feature card, use the **param global password** command in application parameter configuration mode. To restore the default value, use the **no** form of this command.

**param global-password** *password*
**no param global-password** *password*

## Syntax Description

| | |
|---|---|
| *password* | Character string used to define the CiscoSecure for Windows NT password to be used with store-and-forward fax. The maximum length is 64 alphanumeric characters. |

## Command Default

No password is defined

## Command Modes

Application parameter configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command is introduced to replace the **call application voice global-password** command. |

## Usage Guidelines

CiscoSecure for Windows NT might require a separate password to complete authentication, no matter what security protocol you use. This command defines the password to be used with CiscoSecure for Windows NT. All records on the Windows NT server use this defined password.

This command applies to on-ramp store-and-forward fax functions on Cisco AS5300 universal access server voice feature cards. It is not used on modem cards.

## Examples

The following example shows a password (abercrombie) being used by AAA for the app_libretto_onramp9 Tcl application:

```
application
service onramp tftp://sample/tftp-users/tcl/app_libretto_onramp9.tcl
param global-password abercrombie
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice global-password** | Defines a password to be used with CiscoSecure for Windows NT when using store-and-forward fax on a voice feature card. |

# param language

To configure the language parameter in a service or package on the gateway, use the **param language**command in application parameter configuration mode. There is no **no** form of this command.

**param language** *prefix*

## Syntax Description

| | |
|---|---|
| *prefix* | Two-character prefix for the language; for example, "*en"* for English or "*ru"* for Russian. |

## Command Default

No default behavior or values

## Command Modes

Application parameter configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call language voice** command. |

## Usage Guidelines

Before you configure the language parameter, you must load the language package using the **package language** command in application configuration mode.

If you are configuring this parameter for a service, use the following commands:

**service** *name url*

**param language** *prefix*

## Examples

The following example adds Russian (*ru* ) as a Tcl module and configures the debitcard application to use Russian for prompts:

```
application
package language ru tftp://box/unix/scripts/multi-lang/ru_translate.tcl
service debitcard tftp://server-1/tftpboot/scripts/app_debitcard.2.0.2.8.tcl
param language ru
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice set-location** | Defines the category and location of audio files that are used for dynamic prompts by the specified IVR application (Tcl or VoiceXML). |
| **call language voice** | Configures an external Tcl module for use with an IVR application. |
| **param** | Loads and configures parameters in a package or a service (application). |
| **param account-id-method** | Configures an application to use a particular method to assign the account identifier. |
| **param convert-discpi-after-connect** | Enables or disables conversion of a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |

| Command | Description |
|---|---|
| **param event-log** | Enables or disables logging for linkable Tcl functions (packages). |
| **param mode** | Configures the call transfer mode for a package. |
| **param pin-len** | Defines the number of characters in the PIN for an application. |
| **param redirect-number** | Defines the telephone number to which a call is redirected--for example, the operator telephone number of the service provider--for an application. |
| **param reroutemode** | Configures the call transfer reroutemode (call forwarding) for a package. |
| **param retry-count** | Defines the number of times a caller is permitted to reenter the PIN for a designated application and passes that information to the application. |
| **param security** | Configures security for linkable Tcl functions (packages). |
| **param uid-length** | Defines the number of characters in the UID for a package. |
| **param warning-time** | Defines the number of seconds of warning that a user receives before the allowed calling time expires. |

# param mail-script

To specify the VoiceXML application to which the off-ramp mail application hands off a call when the destination telephone answers, use the **param mail-script**command in application parameter configuration mode. To remove the application, use the **no** form of this command.

**param mail-script** *application-name*
**no param mail-script** *application-name*

## Syntax Description

| | |
|---|---|
| *application-name* | Name of the VoiceXML application to which the off-ramp mail application hands off the call when the destination answers. |

## Command Default

No default behavior or values

## Command Modes

Application parameter configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command is introduced to replace the **call application voice mail-script** command. |

## Usage Guidelines

- To configure the mail application onto the gateway, use the **application** command.
- The off-ramp mail application must be configured in the Multimedia Mail over Internet Protocol (MMoIP) dial peer that matches the telephone number contained in the header of the incoming e-mail message.
- The off-ramp mail application must use the Tool Command Language (Tcl) script named "app_voicemail_offramp.tcl" that is provided by Cisco. You can download this Tcl script from the Cisco website by following this path:

Cisco.com > Technical Support & Documentation > Tools & Resources > Software Downloads > Access Software > TclWare

## Examples

The following example shows that the off-ramp mail application named "offramp-mapp" hands calls to the application named "mapp-test" if the telephone number in the e-mail header is seven digits beginning with 555 :

```
application
service offramp-mapp tftp://sample/tftp-users/tcl/app_voicemail_offramp.tcl
param mail-script mapp-test
!
dial-peer voice 1001 mmoip
 application offramp-mapp
 incoming called-number 555....
 information-type voice
```

**Related Commands**

| Command | Description |
|---|---|
| **call application voice mail-script** | Specifies the VoiceXML application to which the off-ramp mail application hands off a call when the destination telephone answers. |

# param mode

To configure the call transfer mode for a package, use the **param mode** command in application parameter configuration mode. To reset to the default, use the **no** form of this command.

**param mode** {**redirect** | **redirect-at-alert** | **redirect-at-connect** | **redirect-rotary** | **rotary**}
**no param mode**

## Syntax Description

| | |
|---|---|
| **redirect** | Gateway redirects the call leg to the redirected destination number. |
| **redirect-at-alert** | Gateway places a new call to the redirected destination number and initiates a call transfer when the outgoing call leg is in the alert state. If the call transfer is successful, the two call legs are disconnected on the gateway. If the transfer fails, the gateway bridges the two call legs. Supports Two B-Channel Transfer (TBCT). |
| **redirect-at-connect** | Gateway places a new call to the redirected destination number and initiates a call transfer when the outgoing call leg is in the connect state. If the call transfer is successful, the two call legs are disconnected on the gateway. If the transfer fails, the gateway bridges the two call legs. Supports TBCT. |
| **redirect-rotary** | Gateway redirects the call leg to the redirected destination number. If redirection fails, the gateway places a rotary call to the redirected destination number and hairpins the two call legs. For TBCT, this mode is the same as **redirect-at-connect**. |
| **rotary** | Gateway places a rotary call for the outgoing call leg and hairpins the two call legs. Call redirection is not invoked. This is the default. |

## Command Default

Rotary method; call redirection is not invoked.

## Command Modes

Application parameter configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced. |

## Usage Guidelines

This command is used to configure call transfer mode for a package only. You can then configure one or more services to use that package. Alternatively, you can use the **paramspace callsetup mode**command to configure call transfer mode for a service, or standalone application.

## Examples

The following example shows the call transfer method set to redirect for the call setup package:

```
application
package callsetup
param mode redirect
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice mode** | Directs the fax detection IVR application to operate in one of its four connection modes. |
| **call application voice transfer mode** | Specifies the call-transfer method for Tcl)or VoiceXML applications. |
| **param** | Loads and configures parameters in a package or a service (application). |
| **param account-id-method** | Configures an application to use a particular method to assign the account identifier. |
| **param convert-discpi-after-connect** | Enables or disables conversion of a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |
| **param event-log** | Enables or disables logging for linkable Tcl functions (packages). |
| **param language** | Configures the language parameter in a service or package on the gateway. |
| **param pin-len** | Defines the number of characters in the personal identification number (PIN) for an application. |
| **param redirect-number** | Defines the telephone number to which a call is redirected--for example, the operator telephone number of the service provider--for an application. |
| **param reroutemode** | Configures the call transfer reroutemode (call forwarding) for a package. |
| **param retry-count** | Defines the number of times a caller is permitted to reenter the PIN for a designated application and passes that information to the application. |
| **param security** | Configures security for linkable Tcl functions (packages). |
| **param uid-length** | Defines the number of characters in the UID for a package. |
| **param warning-time** | Defines the number of seconds of warning that a user receives before the allowed calling time expires. |

# param pin-len

To define the number of characters in the personal identification number (PIN) for an application, use the **param pin len**command in application parameter configuration mode. To disable the PIN for the designated application, use the no form of this command.

**param pin-len** *number*
**no param pin-len** *number*

## Syntax Description

| | |
|---|---|
| *number* | Number of allowable characters in PINs associated with the specified application. Range is from 0 to 10. The default is 4. |

## Command Default

No default behavior or values

## Command Modes

Application parameter configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice pin-len** command. |

## Usage Guidelines

Use this command when configuring interactive voice response (IVR)--depending on the Tool Command Language (Tcl) script being used--or one of the IVR-related features (such as Debit Card) to define the number of allowable characters in a PIN for the specified application and to pass that information to the specified application.

To configure the PIN length for a package, load the package using the **package** command before using the **param pin-len** command. To configure the PIN length for a service, use the **service** command before using the **param pin-len** command.

## Examples

The following example shows how to define a PIN length of 8 characters for a Tcl digit collection package:

```
application
package digcl.tcl
param pin-len 8
```

The following example shows how to define a PIN length of 8 characters for a debit card application:

```
application
service debitcard tftp://tftp-server/dc/app_debitcard.tcl
param pin-len 8
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice pin-len** | Defines the number of characters in the PIN for the designated application. |
| **param** | Loads and configures parameters in a package or a service (application). |

| Command | Description |
|---|---|
| **param account-id-method** | Configures an application to use a particular method to assign the account identifier. |
| **param convert-discpi-after-connect** | Enables or disables conversion of a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |
| **param event-log** | Enables or disables logging for linkable Tcl functions (packages). |
| **param language** | Configures the language parameter in a service or package on the gateway. |
| **param mode** | Configures the call transfer mode for a package. |
| **param redirect-number** | Defines the telephone number to which a call is redirected--for example, the operator telephone number of the service provider--for an application. |
| **param reroutemode** | Configures the call transfer reroutemode (call forwarding) for a package. |
| **param retry-count** | Defines the number of times a caller is permitted to reenter the PIN for a designated application and passes that information to the application. |
| **param security** | Configures security for linkable Tcl functions (packages). |
| **param uid-length** | Defines the number of characters in the UID for a package. |
| **param warning-time** | Defines the number of seconds of warning that a user receives before the allowed calling time expires. |

# param prompt

To direct the fax detection interactive voice response (IVR) application to use the specified audio file as a user prompt, use the **param prompt** command in application parameter configuration mode. To disable use of this audio file, use the **no** form of this command.

**param prompt** *prompt-url*
**no param prompt** *prompt-url*

## Syntax Description

| | |
|---|---|
| *prompt-url* | The URL or Cisco IOS file system (IFS) location on the TFTP server for the audio file containing the prompt for the application. |

## Command Default

The prompt space is empty and no prompt is played.

## Command Modes

Application parameter configuration (config-app-param)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE 26.2.1 | This command was modified to display warning message for insecure protocol usage. |
| 12.3(14)T | This command is introduced to replace the **call application voice prompt** command. |

## Usage Guidelines

This command is useful only in the listen-first, default-voice, and default-fax modes of the fax detection application.

Audio files should be a minimum of 9 seconds long so that callers do not hear silence during the initial CNG detection period. Any .au file can be used; formats are described in the Cisco IOS Voice, Video, and Fax Configuration Guide, Release 12.4.

**Warning** Starting with Cisco IOS XE 26.2.1, a warning message is displayed when insecure protocols such as HTTP, FTP, or TFTP are used, as they do not provide encryption or authentication. Use secure alternatives such as HTTPS, SFTP, or SCP instead.

## Examples

The following example associates the audio file" promptfile.au" with the application file "fax_detect", and the application with the inbound POTS dial peer:

```
application
service fax_detect tftp://users/scripts/app_fax_detect.2.x.x.tcl
param mode default-voice
param prompt promptfile.au
dial-peer voice 302 pots
 application fax_detect
```

The following example shows the warning message displayed for the usage of insecure HTTP, TFTP, or FTP protocol:

```
Device(config)#application

Device(config-app)#service fax_detect tftp://users/scripts/app_fax_detect.2.x.x.tcl
SECURITY WARNING - Module: CONFIG_VOICE, Command: service fax_detect
tftp://users/scripts/app_fax_detect.2.x.x.tcl,
Reason: Legacy protocol poses data confidentiality and integrity risks due to lack of
encryption and authentication,
Description: service debitcard configured with FTP/HTTP/TFTP URL protocol,
Remediation: Use secure protocols such as HTTPS, SFTP, or SCP instead of insecure HTTP,
FTP, or TFTP

Device(config-app-param)#http://10.10.10.70//tftpboot/COLT-DE-CUCM.cfg
SECURITY WARNING - Module: VOICE_APP, Command: param prompt
http://10.10.10.70//tftpboot/COLT-DE-CUCM.cfg,
Reason: Legacy protocol poses data confidentiality and integrity risks due to lack of
encryption and authentication,
Description: param prompt configured with FTP/HTTP/TFTP URL protocol,
Remediation: Use secure protocols such as HTTPS, SFTP, or SCP instead of insecure HTTP,
FTP, or TFTP

Device(config-app-param)#
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice prompt** | Directs the fax detection interactive voice response (IVR) application to use the specified audio file as a user prompt. |

# param redirect-number

To define the telephone number to which a call is redirected--for example, the operator telephone number of the service provider--for an application, use the **param redirect number** command in application parameter configuration mode. To cancel the redirect telephone number, use the **no** form of this command.

**param redirect-number** *number*
**no param redirect-number** *number*

## Syntax Description

| | |
|---|---|
| *number* | Designated operator telephone number of the service provider (or any other number designated by the customer). This is the number where calls are terminated when, for example, allowed debit time has run out or the debit amount is exceeded. |

## Command Default

No default behavior or values

## Command Modes

Application parameter configuration

## Command History

| Cisco IOS Release | Cisco Product | Modification |
|---|---|---|
| 12.3(14)T | Cisco CME 3.3 | This command was introduced to replace the **call application voice redirect-number** command. |

## Usage Guidelines

Use this command when configuring interactive voice response (IVR)--depending on the Tool Command Language (Tcl) script being used--or one of the IVR-related features (such as Debit Card) to define the telephone number to which a call is redirected.

To configure the redirect number for a package, load the package using the **package** command before using the **param redirect-number**command. To configure the redirect number for a service, use the **service** command before using the **param redirect-number**command.

## Examples

The following example shows how to define a redirect number for the application named "prepaid":

```
application
service prepaid tftp://tftp-server/scripts/prepaid.tcl
param redirect-number 5550111
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice redirect-number** | Defines the telephone number to which a call is redirected--for example, the operator telephone number of the service provider--for the designated application. |
| **param** | Loads and configures parameters in a package or a service (application). |
| **param account-id-method** | Configures an application to use a particular method to assign the account identifier. |

| Command | Description |
|---|---|
| **param convert-discpi-after-connect** | Enables or disables conversion of a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |
| **param event-log** | Enables or disables logging for linkable Tcl functions (packages). |
| **param language** | Configures the language parameter in a service or package on the gateway. |
| **param mode** | Configures the call transfer mode for a package. |
| **param pin-len** | Defines the number of characters in the personal identification number (PIN) for an application. |
| **param reroutemode** | Configures the call transfer reroutemode (call forwarding) for a package. |
| **param retry-count** | Defines the number of times a caller is permitted to reenter the PIN for a designated application and passes that information to the application. |
| **param security** | Configures security for linkable Tcl functions (packages). |
| **param uid-length** | Defines the number of characters in the UID for a package. |
| **param warning-time** | Defines the number of seconds of warning that a user receives before the allowed calling time expires. |
| **service** | Loads and configures a specific, standalone application on a dial peer. |

# param reroutemode

To configure the call transfer reroutemode (call forwarding) for a package, use the **param reroutemode** command in application parameter configuration mode. To reset to the default, use the **no** form of this command.

**param reroutemode** {**redirect** | **redirect-at-alert** | **redirect-at-connect** | **redirect-rotary** | **rotary**}
**no param reroutemode**

## Syntax Description

| | |
|---|---|
| **redirect** | Two call legs are directly connected. Supports RTPvt. |
| **redirect-at-alert** | Gateway places a new call to the redirected destination number and initiates a call transfer when the outgoing call leg is in the alert state. If the call transfer is successful, the two call legs are disconnected on the gateway. If the transfer fails, the gateway bridges the two call legs. Supports Two B-Channel Transfer (TBCT). |
| **redirect-at-connect** | Gateway places a new call to the redirected destination number and initiates a call transfer when the outgoing call leg is in the connect state. If the call transfer is successful, the two call legs are disconnected on the gateway. If the transfer fails, the gateway bridges the two call legs. Supports TBCT. |
| **redirect-rotary** | Two call legs are directly connected (redirect). If that fails, the two call legs are hairpinned on the gateway (rotary). |
| **rotary** | Gateway places a rotary call for the outgoing call leg and hairpins the two calls together. Release-to-Pivot (RTPvt) is not invoked. This is the default. |

## Command Default

Rotary method; RTPvt is not invoked.

## Command Modes

Application parameter configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced. |

## Usage Guidelines

This command is used to configure call forwarding for a package only. You can then configure one or more services to use that package. Alternatively, you can use the **paramspace callsetup reroutemode**command to configure call forwarding for a service, or standalone application.

Redirect-rotary is the preferred transfer method because it ensures that a call-redirect method is always selected, provided that the call leg is capable of it.

## Examples

The following example shows the call forwarding method set to redirect for the call setup package:

```
application
package callsetup
param reroutemode redirect
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice transfer reroute-mode** | Specifies the call-forwarding behavior of a Tcl application. |
| **param** | Loads and configures parameters in a package or a service (application). |
| **param account-id-method** | Configures an application to use a particular method to assign the account identifier. |
| **param convert-discpi-after-connect** | Enables or disables conversion of a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |
| **param event-log** | Enables or disables logging for linkable Tcl functions (packages). |
| **param language** | Configures the language parameter in a service or package on the gateway. |
| **param mode** | Configures the call transfer mode for a package. |
| **param pin-len** | Defines the number of characters in the PIN for an application. |
| **param redirect-number** | Defines the telephone number to which a call is redirected--for example, the operator telephone number of the service provider--for an application. |
| **param retry-count** | Defines the number of times a caller is permitted to reenter the PIN for a designated application and passes that information to the application. |
| **param security** | Configures security for linkable Tcl functions (packages). |
| **param uid-length** | Defines the number of characters in the UID for a package. |
| **param warning-time** | Defines the number of seconds of warning that a user receives before the allowed calling time expires. |

# param retry-count

To define the number of times that a caller is permitted to reenter the personal identification number (PIN) for a package, use the **param retry count** command in application parameter configuration mode. To cancel the configured retry count, use the **no** form of this command.

**param retry-count** *number*
**no param retry-count** *number*

## Syntax Description

| | |
|---|---|
| *number* | Number of times the caller is permitted to reenter PIN digits. Range is 1 to 5. The default is 3. |

## Command Default

3

## Command Modes

Application parameter configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced. |

## Usage Guidelines

Use this command when configuring interactive voice response (IVR)--depending on the Tool Command Language (Tcl) script being used--or one of the IVR-related features (such as Debit Card) to define how many times a user can reenter a PIN.

To configure the PIN retry count for a package, load the package using the **package** command before using the **param retry-count** command. To configure the PIN retry count for a service, use the **service** command before using the **param retry-count**command.

## Examples

The following example shows how to configure the PIN retry count in a package so that a user can reenter a PIN two times before being disconnected.

```
application
package sample1.tcl
param retry-count 2
```

The following example shows how to configure the PIN retry count in a debit card application so that a user can reenter a PIN two times before being disconnected.

```
application
service debitcard tftp://tftp-server/dc/app_debitcard.tcl
param retry-count 2
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice retry-count** | Defines the number of times that a caller is permitted to reenter the PIN for the designated application. |
| **param** | Loads and configures parameters in a package or a service (application). |

| Command | Description |
|---|---|
| **param account-id-method** | Configures an application to use a particular method to assign the account identifier. |
| **param convert-discpi-after-connect** | Enables or disables conversion of a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |
| **param event-log** | Enables or disables logging for linkable Tcl functions (packages). |
| **param language** | Configures the language parameter in a service or package on the gateway. |
| **param mode** | Configures the call transfer mode for a package. |
| **param pin-len** | Defines the number of characters in the PIN for an application. |
| **param redirect-number** | Defines the telephone number to which a call is redirected--for example, the operator telephone number of the service provider--for an application. |
| **param reroutemode** | Configures the call transfer reroutemode (call forwarding) for a package. |
| **param security** | Configures security for linkable Tcl functions (packages). |
| **param uid-length** | Defines the number of characters in the UID for a package. |
| **param warning-time** | Defines the number of seconds of warning that a user receives before the allowed calling time expires. |

# param security

To configure security for linkable Tcl functions (packages), use the **param security** command in application parameter configuration mode. To restore this parameter to the default value, use the **no** form of this command.

**param security** {**trusted** | **untrusted**}
**no param security** {**trusted** | **untrusted**}

## Syntax Description

| | |
|---|---|
| **trusted** | Automatic number identification (ANI) is not blocked. |
| **untrusted** | ANI is blocked. |

## Command Default

No default behavior or values

## Command Modes

Application parameter configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice security** command. |

## Usage Guidelines

This command is available for the built-in common voice application package. If you are configuring this parameter for that package, you must first use the command **package appcommon**.

If you are configuring this parameter for a service, use the following commands:

**service** *name url*

**paramspace appcommon security** {**trusted** | **untrusted**}

If an application is configured as a trusted application, it is trusted not to provide the calling number to the destination party, so ANI is always provided if available. Normally, the voice gateway does not provide the calling number (ANI) to a VoiceXML application if the caller ID is blocked. Caller ID is blocked if a call that comes into the voice gateway has the presentation indication field set to "presentation restricted". The session.telephone.ani variable is set to "blocked". When the **param security trusted** command is configured, the gateway does not block caller ID; it provides the calling number to the VoiceXML application. If the keyword of this command is set to untrusted, caller ID is blocked.

To enable GTD (Generic Transparency Descriptor) parameters in call signaling messages to map to VoiceXML and Tcl session variables, the **param securit y trusted** command must be configured. If this command is not configured, the VoiceXML variables that correspond to GTD parameters are marked as not available. For a detailed description of the VoiceXML and Tcl session variables, see the Cisco VoiceXML Programmer's Guide and the Tcl IVR API Version 2.0 Programmer's Guide, respectively.

## Examples

The following example shows using the **param security trusted** command to set the security level of the common application package to "trusted" so that automatic number identification (ANI) is not blocked.

```
application
package appcommon
param security trusted
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice security trusted** | Sets the security level of a VoiceXML application to "trusted" so that ANI is not blocked. |
| **package appcommon** | Configures parameters in the built-in common voice application package. |
| **param** | Loads and configures parameters in a package or a service (application). |
| **param account-id-method** | Configures an application to use a particular method to assign the account identifier. |
| **param convert-discpi-after-connect** | Enables or disables conversion of a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |
| **param event-log** | Enables or disables logging for linkable Tcl functions (packages). |
| **param language** | Configures the language parameter in a service or package on the gateway. |
| **param mode** | Configures the call transfer mode for a package. |
| **param pin-len** | Defines the number of characters in the PIN for an application. |
| **param redirect-number** | Defines the telephone number to which a call is redirected--for example, the operator telephone number of the service provider--for an application. |
| **param reroutemode** | Configures the call transfer reroutemode (call forwarding) for a package. |
| **param retry-count** | Defines the number of times a caller is permitted to reenter the PIN for a designated application and passes that information to the application. |
| **paramspace appcommon security** | Configures security for a service (application). |
| **param uid-length** | Defines the number of characters in the UID for a package. |
| **param warning-time** | Defines the number of seconds of warning that a user receives before the allowed calling time expires. |
| **service** | Loads and configures a specific, standalone application on a dial peer. |

# param uid-len

To define the number of characters in the user identification number (UID) for a package, use the **param uid-len**command in application parameter configuration mode. To restore the default setting for this command, use the **no** form of this command.

**param uid-len** *number*
**no param uid-len** *number*

## Syntax Description

| | |
|---|---|
| *number* | Number of allowable characters in UIDs that are associated with the specified application. Range is from 1 to 20. Default is 10. |

## Command Default

10 characters

## Command Modes

Application parameter configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice uid-length** command. |

## Usage Guidelines

Use this command when configuring interactive voice response (IVR)--depending on the Tool Command Language (Tcl) script being used--or one of the IVR-related features (such as Debit Card) to define the number of allowable characters in a UID.

This command is available for the built-in common voice application package. If you are configuring this parameter for that package, you must first use the command **package appcommon**. If you are configuring this parameter for a service, you must first use the **service** command

## Examples

The following example configures the UID length to 20 in a package.

```
application
package sample1.tcl
param uid-len 20
```

The following example configures the UID length to 20 in a debit-card application.

```
application
service debitcard tftp://tftp-server/dc/app_debitcard.tcl
param uid-len 20
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice uid-length** | Defines the number of characters in the UID for the designated application and to pass that information to the specified application. |
| **package appcommon** | Configures parameters in the built-in common voice application package. |

| Command | Description |
|---|---|
| **param** | Loads and configures parameters in a package or a service (application). |
| **param account-id-method** | Configures an application to use a particular method to assign the account identifier. |
| **param convert-discpi-after-connect** | Enables or disables conversion of a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |
| **param event-log** | Enables or disables logging for linkable Tcl functions (packages). |
| **param language** | Configures the language parameter in a service or package on the gateway. |
| **param mode** | Configures the call transfer mode for a package. |
| **param pin-len** | Defines the number of characters in the PIN for an application. |
| **param redirect-number** | Defines the telephone number to which a call is redirected--for example, the operator telephone number of the service provider--for an application. |
| **param reroutemode** | Configures the call transfer reroutemode (call forwarding) for a package. |
| **param retry-count** | Defines the number of times a caller is permitted to reenter the PIN for a designated application and passes that information to the application. |
| **param security** | Configures security for linkable Tcl functions (packages). |
| **param warning-time** | Defines the number of seconds of warning that a user receives before the allowed calling time expires. |

# param voice-dtmf

To direct the fax detection interactive voice response (IVR) application to recognize a specified digit to indicate a voice call, use the **param voice dtmf** command in application parameter configuration mode. To remove configuration of this digit, use the **no** form of this command.

**param voice-dtmf** {**0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | ***** | **#**}
**no param voice-dtmf** {**0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | ***** | **#**}

## Syntax Description

| | |
|---|---|
| **0** \| **1**\| **2**\| **3**\| **4**\| **5**\| **6**\| **7**\| **8**\| **9**\| *****\| **#** | The telephone keypad button pressed by the calling party to indicate a voice call, in response to the audio prompt configured in default-voice and default-fax mode of the fax detection IVR application. |

## Command Default

**1**

## Command Modes

Application parameter configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command is introduced to replace the **call application voice voice-dtmf**command. |

## Usage Guidelines

This command is useful only when the fax detection IVR application is being configured in default-voice mode or default-fax mode, as defined by the **param mode** command.

If you also configure voice DTMF using the **param voice-dtmf** command, you must use different numbers for the voice and fax DTMF digits.

## Examples

The following example selects digit 2 Dual tone multifrequency (DTMF) to indicate a voice call:

```
application
service faxdetect tftp://sample/tftp-users/tcl/app_fax_detect.2.x.x.tcl
param voice-dtmf 2
dial-peer voice 302 pots
 application fax_detect
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice voice-dtmf** | Directs the fax detection IVR application to recognize a specified digit to indicate a voice call. |
| **param mode** | Configures the call transfer mode for a package. |
| **param fax-dtmf** | Directs an application to recognize a specified digit to indicate a fax call in default-voice and default-fax modes. |

# param warning-time

To define the number of seconds of warning that a user receives before the allowed calling time expires use the **param warning time** command in application parameter configuration mode. To remove the configured warning period, use the **no** form of this command.

**param warning-time** *number*
**no param warning-time** *number*

## Syntax Description

| | |
|---|---|
| *number* | Length of the warning period, in seconds, before the allowed calling time expires. Range is from 10 to 600. This argument has no default value. |

## Command Default

No default behavior or values

## Command Modes

Application parameter configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice warning-time** command. |

## Usage Guidelines

Use this command when configuring interactive voice response (IVR)--depending on the Tool Command Language (Tcl) script being used--or one of the IVR-related features (such as Debit Card) to define the number of seconds in the warning period before the allowed calling time expires.

This command is available for the built-in common voice application package. If you are configuring this parameter for that package, you must first use the command **package appcommon**. If you are configuring this parameter for a service, you must first use the **service** command

## Examples

The following example configures the warning time parameter to 30 seconds in a package.

```
application
package sample1.tcl
param warning-time 30
```

The following example configures the warning time parameter to 30 seconds in a debit-card application.

```
application
service debitcard tftp://tftp-server/dc/app_debitcard.tcl
param warning-time 30
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice warning-time** | Defines the number of seconds of warning that a user receives before the allowed calling time expires. |
| **package appcommon** | Configures parameters in the built-in common voice application package. |

| Command | Description |
|---|---|
| **param** | Loads and configures parameters in a package or a service (application). |
| **param account-id-method** | Configures an application to use a particular method to assign the account identifier. |
| **param convert-discpi-after-connect** | Enables or disables conversion of a DISCONNECT message with Progress Indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |
| **param event-log** | Enables or disables logging for linkable Tcl functions (packages). |
| **param language** | Configures the language parameter in a service or package on the gateway. |
| **param mode** | Configures the call transfer mode for a package. |
| **param pin-len** | Defines the number of characters in the PIN for an application. |
| **param redirect-number** | Defines the telephone number to which a call is redirected--for example, the operator telephone number of the service provider--for an application. |
| **param reroutemode** | Configures the call transfer reroutemode (call forwarding) for a package. |
| **param retry-count** | Defines the number of times a caller is permitted to reenter the PIN for a designated application and passes that information to the application. |
| **param security** | Configures security for linkable Tcl functions (packages). |
| **param uid-length** | Defines the number of characters in theUID for a package. |
| **service** | Loads and configures a specific, standalone application on a dial peer. |

# paramspace

To enable an application to use parameters from the local parameter space of another application, use the **paramspace** command in application service configuration mode. To return to the default parameter namespace for this parameter, use the **no** form of this command.

**paramspace** *parameter-namespace parameter-name parameter-value*
**no paramspace** *parameter-namespace parameter-name parameter-value*

## Syntax Description

| | |
|---|---|
| *parameter-namespace* | Namespace of the parameter from which you want to use parameters. |
| *parameter-name* | Parameter to use. |
| *parameter-value* | Value of the parameter. |

## Command Default

No default behavior or values

## Command Modes

Application service configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced. |

## Usage Guidelines

To avoid problems with applications using the same parameter names, the *parameter namespace,* or *parameterspace* concept is provided. When an application is defined on the gateway, its parameter namespace is automatically defined. This is known as the application's s local parameterspace. When you use the **param** command to configure an application's parameters, the parameters available for configuration are those contained in the local parameterspace.

If you want to use parameter definitions found in different parameterspace, you can use the **paramspace***parameter-namespaceparameter-name parameter-value*command to map the application's parameters to a different parameterspace. This allows that application to use the parameter definitions found in the new parameterspace, in addition to its local parameterspace.

## Examples

The following example shows a debit card service configured to use parameters from an English language translation package:

```
application
service debitcard tftp://server-1//tftpboot/scripts/app_debitcard.2.0.2.8.tcl
paramspace english language en
  paramspace english index 1
  paramspace english prefix en
  paramspace english location tftp://server-1//tftpboot/scripts/au/en/
```

## Related Commands

| Command | Description |
|---|---|
| **param** | Loads and configures parameters in a package or a service (application) on the gateway. |

| Command | Description |
|---|---|
| **paramspace appcommon event-log** | Enables or disables logging for a service (application). |
| **paramspace appcommon security** | Configures security for a service (application). |
| **paramspace callsetup mode** | Configures the call transfer mode for an application. |
| **paramspace callsetup reroutemode** | Configures the call reroute mode (call forwarding) for an application. |
| **paramspace language** | Defines the category and location of audio files that are used for dynamic prompts by an IVR application (Tcl or VoiceXML). |

# paramspace appcommon event-log

To enable or disable logging for a service (application), use the **paramspace appcommon event-log** command in application service configuration mode. There is no **no** form of this command.

**paramspace appcommon event-log** {**enable** | **disable**}

## Syntax Description

| | |
|---|---|
| **enable** | Event logging is enabled. |
| **disable** | Event logging is disabled. |

## Command Default

No default behavior or values

## Command Modes

Application service configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice event-log** command. |

## Usage Guidelines

Use this command to configure event logging for a service (application).

If you are configuring event logging for a package only, use the **package appcommon** command in application-parameter configuration mode.

If you are configuring event logging for all voice applications, use the **event-log** command in application-configuration monitor mode.

**Note** To prevent event logging from adversely impacting system resources for production traffic, the gateway uses a throttling mechanism. When free processor memory drops below 20%, the gateway automatically disables all event logging. It resumes event logging when free memory rises above 30%. While throttling is occurring, the gateway does not capture any new event logs even if event logging is enabled. You should monitor free memory and enable event logging only when necessary for isolating faults.

## Examples

The following example shows event-logging disabled for a debit-card application.

```
application
service debitcard tftp://tftp-server/dc/app_debitcard.tcl
paramspace appcommon event-log disable
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice event-log** | Enables event logging for a specific voice application. |
| **paramspace** | Enables an application to use parameters from the local parameter space of another application. |

| Command | Description |
|---|---|
| **paramspace appcommon security** | Configures security for a service (application). |
| **paramspace callsetup mode** | Configures the call transfer mode for an application. |
| **paramspace callsetup reroutemode** | Configures the call reroute mode (call forwarding) for an application. |
| **paramspace language** | Defines the category and location of audio files that are used for dynamic prompts by an IVR application (Tcl or VoiceXML). |

# paramspace appcommon security

To configure security for a service (application), use the **paramspace appcommon security** command in application service configuration mode. To return to the default parameter namespace for this parameter, use the **no** form of this command.

**paramspace appcommon security** {**trusted** | **untrusted**}
**no paramspace appcommon security** {**trusted** | **untrusted**}

## Syntax Description

| | |
|---|---|
| **trusted** | Automatic number identification (ANI) is not blocked. |
| **untrusted** | ANI is blocked. |

## Command Default

No default behavior or values

## Command Modes

Application service configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice security** command. |

## Usage Guidelines

This command is available for the built-in common voice application package. If you are configuring this parameter for the built-in common voice application package, use the command **param security** command.

If an application is configured as a trusted application, it is trusted not to provide the calling number to the destination party, so ANI is always provided if available. Normally, the voice gateway does not provide the calling number (ANI) to a VoiceXML application if the caller ID is blocked. Caller ID is blocked if a call that comes into the voice gateway has the presentation indication field set to "presentation restricted". The session.telephone.ani variable is set to "blocked". When the **paramspace appcommon security trusted**command is configured, the gateway does not block caller ID; it provides the calling number to the VoiceXML application. If the keyword of this command is set to untrusted, caller ID is blocked.

To enable GTD (Generic Transparency Descriptor) parameters in call signaling messages to map to VoiceXML and Tcl session variables, the **paramspace appcommon security trusted** command must be configured. If this command is not configured, the VoiceXML variables that correspond to GTD parameters are marked as not available. For a detailed description of the VoiceXML and Tcl session variables, see the Cisco VoiceXML Programmer's Guide and the Tcl IVR API Version 2.0 Programmer's Guide, respectively.

## Examples

The following example shows security configured for a debit card application. The security level of the application is set to "trusted" so that automatic number identification (ANI) is not blocked.

```
application
service debitcard tftp://tftp-server/dc/app_debitcard.tcl
paramspace appcommon security trusted
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice security trusted** | Sets the security level of a VoiceXML application to "trusted" so that ANI is not blocked. |
| **paramspace** | Enables an application to use parameters from the local parameter space of another application. |
| **paramspace appcommon event-log** | Enables or disables logging for a service (application). |
| **paramspace callsetup mode** | Configures the call transfer mode for an application. |
| **paramspace callsetup reroutemode** | Configures the call reroute mode (call forwarding) for an application. |
| **paramspace language** | Defines the category and location of audio files that are used for dynamic prompts by an IVR application (Tcl or VoiceXML). |

# paramspace callsetup mode

To configure the call transfer mode for an application, use the **paramspace callsetup mode** command in application service configuration mode. To reset to the default, use the **no** form of this command.

**paramspace callsetup mode** {**redirect** | **redirect-at-alert** | **redirect-at-connect** | **redirect-rotary** | **rotary**}
**no paramspace callsetup mode**

## Syntax Description

| | |
|---|---|
| **redirect** | Gateway redirects the call leg to the redirected destination number. |
| **redirect-at-alert** | Gateway places a new call to the redirected destination number and initiates a call transfer when the outgoing call leg is in the alert state. If the call transfer is successful, the two call legs are disconnected on the gateway. If the transfer fails, the gateway bridges the two call legs. Supports Two B-Channel Transfer (TBCT). |
| **redirect-at-connect** | Gateway places a new call to the redirected destination number and initiates a call transfer when the outgoing call leg is in the connect state. If the call transfer is successful, the two call legs are disconnected on the gateway. If the transfer fails, the gateway bridges the two call legs. Supports TBCT. |
| **redirect-rotary** | Gateway redirects the call leg to the redirected destination number. If redirection fails, the gateway places a rotary call to the redirected destination number and hairpins the two call legs. For TBCT, this mode is the same as **redirect-at-connect**. |
| **rotary** | Gateway places a rotary call for the outgoing call leg and hairpins the two call legs. Call redirection is not invoked. This is the default. |

## Command Default

Rotary method; call redirection is not invoked.

## Command Modes

Application service configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice transfer mode** command. |

## Usage Guidelines

Use this command to configure the call transfer mode for a service, or standalone application. Alternatively, you can use the **package callsetup**and **param mode** commands to configure call transfer mode for a package only, and then configure one or more services to use that package.

This command determines whether a voice application can invoke TBCT or RTPvt.

Redirect-rotary is the preferred transfer method because it ensures that a call-redirect method is always selected if the call leg is capable of it.

## Examples

The following example shows the call method set to redirect for a debit-card application:

```
application
service debitcard tftp://tftp-server/dc/app_debitcard.tcl
paramspace callsetup mode redirect
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice transfer mode** | Specifies the call-transfer method for Tcl)or VoiceXML applications. |
| **package callsetup** | Configures parameters in the built-in call-setup package. |
| **param mode** | Configures the call-transfer mode for a package. |
| **paramspace** | Enables an application to use parameters from the local parameter space of another application. |
| **paramspace appcommon event-log** | Enables or disables logging for a service (application). |
| **paramspace appcommon security** | Configures security for a service (application). |
| **paramspace callsetup reroutemode** | Configures the call reroute mode (call forwarding) for an application. |
| **paramspace language** | Defines the category and location of audio files that are used for dynamic prompts by an IVR application (Tcl or VoiceXML). |

# paramspace callsetup reroutemode

To configure the call reroute mode (call forwarding) for an application, use the **paramspace callsetup reroutemode** command in application service configuration mode. To reset to the default, use the **no** form of this command.

**paramspace callsetup reroutemode** {**redirect** | **redirect-at-alert** | **redirect-at-connect** | **redirect-rotary** | **rotary**}
**no paramspace callsetup reroutemode**

## Syntax Description

| | |
|---|---|
| **redirect** | Gateway redirects the call leg to the redirected destination number. |
| **redirect-at-alert** | Gateway places a new call to the redirected destination number and initiates a call transfer when the outgoing call leg is in the alert state. If the call transfer is successful, the two call legs are disconnected on the gateway. If the transfer fails, the gateway bridges the two call legs. Supports Two B-Channel Transfer (TBCT). |
| **redirect-at-connect** | Gateway places a new call to the redirected destination number and initiates a call transfer when the outgoing call leg is in the connect state. If the call transfer is successful, the two call legs are disconnected on the gateway. If the transfer fails, the gateway bridges the two call legs. Supports TBCT. |
| **redirect-rotary** | Gateway redirects the call leg to the redirected destination number. If redirection fails, the gateway places a rotary call to the redirected destination number and hairpins the two call legs. For TBCT, this mode is the same as **redirect-at-connect**. |
| **rotary** | Gateway places a rotary call for the outgoing call leg and hairpins the two call legs. Call redirection is not invoked. This is the default. |

## Command Default

Rotary method; call redirection is not invoked.

## Command Modes

Application service configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice transfer reroute-mode**command. |

## Usage Guidelines

This command is used to configure the call forward mode for a service, or standalone application. Alternatively, you can use the **package callsetup param reroutemode** command to configure call forward mode for a package only, and then configure one or more services to use that package.

This command determines whether a voice application can invoke TBCT or RTPvt.

Redirect-rotary is the preferred transfer method because it ensures that a call-redirect method is always selected if the call leg is capable of it.

## Examples

The following example shows the call forward method set to redirect for a debitcard application:

```
application
service debitcard tftp://tftp-server/dc/app_debitcard.tcl
paramspace callsetup reroutemode redirect
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice transfer reroute-mode** | Specifies the call-forwarding behavior of a Tcl application. |
| **paramspace** | Enables an application to use parameters from the local parameter space of another application. |
| **paramspace appcommon event-log** | Enables or disables logging for a service (application). |
| **paramspace appcommon security** | Configures security for a service (application). |
| **paramspace callsetup mode** | Configures the call transfer mode for an application. |
| **paramspace language** | Defines the category and location of audio files that are used for dynamic prompts by an IVR application (Tcl or VoiceXML). |

# paramspace language

To define the category and location of audio files that are used for dynamic prompts by an IVR application (Tcl or VoiceXML), use the **paramspace language**command in application service configuration mode. To remove these definitions, use the **no** form of this command.

To configure the language parameter in a service or package on the gateway, use the **param language** command in application service configuration mode.

**paramspace** *language* {**location** *location* | **index** *number* | **language** *prefix*}

## Syntax Description

| | |
|---|---|
| *language* | Name of the language package. Cisco IOS software includes some built-in language packages, such as English. |
| **location** *location* | URL of the audio files. Valid URLs refer to TFTP, FTP, HTTP, or RTSP servers, flash memory, or the removable disks on the Cisco 3600 series. |
| **index** *number* | Category group of the audio files (from 0 to 4). For example, audio files representing the days and months can be category 1, audio files representing units of currency can be category 2, and audio files representing units of time--seconds, minutes, and hours--can be category 3. Range is from 0 to 4; 0 means all categories. |
| **language** *prefix* | Two-character code that identifies the language associated with the audio files. Valid entries are as follows:<br>• **en** --English<br>• **sp** --Spanish<br>• **ch** --Mandarin<br>• **aa** --all |

## Command Default

No location, index, or category is set.

## Command Modes

Application service configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice language** and the **call application voice set-location** commands. |

## Usage Guidelines

Tcl scripts and VoiceXML documents can be stored in any of the following locations: On TFTP, FTP, or HTTP servers, in the flash memory on the gateway, or on the removable disks of the Cisco 3600 series. The audio files that they use can be stored in any of these locations, and on RTSP servers.

You can configure multiple set-location lines for a single application.

With the Pre-Paid Debitcard Multi-Language feature, you can create Tcl scripts and a two-character code for any language. See the Cisco Pre-Paid Debitcard Multi-Language Programmer's Reference.

With the multilanguage support for Cisco IOS IVR, you can create a Tcl language module for any language and any set of Text-to-Speech (TTS) notations for use with Tcl and VoiceXML applications. See the Enhanced Multi-Language Support for Cisco IOS Interactive Voice Response document.

## Examples

The following example shows how to configure the **paramspace language**command for a debitcard application.

```
application
service debitcard tftp://server-1//tftpboot/scripts/app_debitcard.2.0.2.8.tcl
paramspace english language en
  paramspace english index 1
  paramspace english prefix en
  paramspace english location tftp://server-1//tftpboot/scripts/au/en/
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice language** | Specifies the language for dynamic prompts used by an IVR application (Tcl or VoiceXML). |
| **call application voice set-location** | Defines the category and location of audio files that are used for dynamic prompts by the specified IVR application (Tcl or VoiceXML). |
| **paramspace** | Enables an application to use parameters from the local parameter space of another application. |
| **paramspace appcommon event-log** | Enables or disables logging for a service (application). |
| **paramspace appcommon security** | Configures security for a service (application). |
| **paramspace callsetup mode** | Configures the call transfer mode for an application. |
| **paramspace callsetup reroutemode** | Configures the call reroute mode (call forwarding) for an application. |

# paramspace session_xwork convert-discpi-after-connect

To enable or disable conversion of a DISCONNECT message with progress indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state, use the **paramspace session_xwork convert-discpi-after-connect** command in application-service configuration mode. To return to the default parameter namespace for this parameter, use the **no** form of this command.

**paramspace session_xwork convert-discpi-after-connect** {**enable** | **disable**}
**no paramspace session_xwork convert-discpi-after-connect** {**enable** | **disable**}

## Syntax Description

| | |
|---|---|
| **enable** | Convert a DISCONNECT message with progress indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |
| **disable** | Revert to a DISCONNECT message with progress indicator set to PROG_INBAND (PI=8) when the call is in the active state. |

## Command Default

Enabled

## Command Modes

Application-service configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced to replace the **call application voice default disc-prog-ind-at-connect** command. |

## Usage Guidelines

This command has no effect if the call is not in the active state. If you are configuring this parameter for a package, use the **package session xwork** command.

## Examples

The following example shows conversion enabled for a DISCONNECT message with progress indicator set to PROG_INBAND (PI=8):

```
application
service callappl.tcl tftp://tftp-server/callappl.tcl
paramspace session_xwork convert-discpi-after-connect enable
```

## Related Commands

| Command | Description |
|---|---|
| **call application voice default disc-prog-ind-at-connect** | Converts a DISCONNECT message with progress indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |
| **package session xwork** | Configures parameters in the built-in session_xwork package. |
| **param convert-discpi-after-connect** | Enables or disables conversion of a DISCONNECT message with progress indicator set to PROG_INBAND (PI=8) to a regular DISCONNECT message when the call is in the active state. |

| Command | Description |
| --- | --- |
| **paramspace** | Enables an application to use parameters from the local parameter space of another application. |

# pass-thru content

To enable the pass-through of Session Description Protocol (SDP) from in-leg to the out-leg, use the **pass-thru content** command either in global VoIP SIP configuration mode or dial-peer configuration mode. To remove a SDP header from a configured pass-through list, use the **no** form of the command.

**pass-thru content[custom-sdp | sdp {mode | system}| unsupp]**
**no pass-thru content[custom-sdp | sdp {mode | system}| unsupp]**

## Syntax Description

| | |
|---|---|
| **custom-sdp** | Enables the pass-through of custom SDP using SIP Profiles. |
| **sdp** | Enables the pass-through of SDP content. |
| **mode** | Enables the pass-through SDP mode. |
| **system** | Specifies that the pass-through configuration use the global sip-ua value. This keyword is available only for the tenant mode to allow it to fallback to the global configurations. |
| **unsupp** | Enables the pass-through of all unsupported content in a SIP message or request. |

## Command Default

Disabled

## Command Modes

SIP configuration (conf-serv-sip)

Dial peer configuration (config-dial-peer)

Voice class tenant configuration (config-class)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS 15.6(1)T, Cisco IOS XE 3.17S | This command was modified to add keyword: **custom-sdp**. |
| 15.6(2)T and IOS XE Denali 16.3.1 | This command was modified to include the keyword: **system**. |
| Cisco IOS XE Cupertino 17.7.1a | Introduced support for YANG models. |

## Examples

The following example shows how to configure pass-through of custom SDP using SIP Profiles peer rules in global VoIP SIP configuration mode:

```
Router(conf-serv-sip)# pass-thru content custom-sdp
```

The following example shows how to configure pass-through of custom SDP using SIP Profiles in dial-peer configuration mode:

```
Router(config-dial-peer)# voice-class sip pass-thru content custom-sdp
```

The following example shows how to configure pass-through of SDP in global VoIP SIP configuration mode:

```
Router(conf-serv-sip)# pass-thru content sdp
```

The following example shows how to configure pass-through of SDP in voice class tenant configuration mode:

```
Router(config-class)# pass-thru content sdp system
```

The following example shows how to configure pass-through of unsupported content types in dial-peer configuration mode:

```
Router(config-dial-peer)# voice-class sip pass-thru content unsupp
```

# pass-thru headers

To enable the pass-through of a list of headers from a globally configured list, use the **pass-thru headers** command either in global VoIP SIP configuration mode or dial peer configuration mode. To remove a header from a configured pass-through list, use the **no** form of the command.

**pass-thru headers [***number* **| unsupp]**
**no pass-thru headers [***number* **| unsupp]**

## Syntax Description

| | |
|---|---|
| *number* | Specifies the sip-hdr-pass-thru list tag number to be linked as global value. Range is from 1 to 10000. |
| **unsupp** | Enables the pass-through of all unsupported headers. |

## Command Default

Disabled

## Command Modes

SIP configuration (conf-serv-sip)

Dial peer configuration (config-dial-peer)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS 15.6(1)T, Cisco IOS XE 3.17S | This command was modified to add keyword: **system** in the dial-peer configuration mode. |
| Cisco IOS XE Bengaluru 17.4.1a | Introduced support for YANG models. |

## Examples

The following example shows how to configure pass-through of unsupported headers in global VoIP SIP configuration mode:

```
Router(conf-serv-sip)# pass-thru headers unsupp
```

The following example shows how to configure pass-through of unsupported headers in dial-peer configuration mode:

```
Router(config-dial-peer)# voice-class sip pass-thru headers unsupp
```

## Related Commands

| Command | Description |
|---|---|
| **pass-thru** | Passes the Session Description Protocol (SDP) transparently from in-leg to the out-leg with no media negotiation. |
| **passthru-hdr-unsupp** | Enables the pass-thru of all unsupported headers. |
| **voice class sip-hdr-passthrulist** | Configures list of headers to be passed through. |

# passthru-hdr

To add a header name to a configured pass-through list, use the **passthru-hdr** command in voice class configuration mode. To remove a header name from a configured pass-through list, use the **no** form of the command.

**passthru-hdr** *header-name*
**no passthru-hdr** [*header-name*]

## Syntax Description

| | |
|---|---|
| *header-name* | Header name of header to be added in the configured pass-through list. |

## Command Default

No header name is added to the configured pass-through list.

## Command Modes

Voice class configuration mode (config-class)

## Command History

| Release | Modification |
|---|---|
| 15.4(1)T | This command was introduced. |
| Cisco IOS XE Bengaluru 17.4.1a | Introduced support for YANG models. |

## Usage Guidelines

A pass-through list using the **voice class sip-hdr-passthrulist** command must be configured before adding a header name to the list.

You can configure a list of headers to be passed through. The list can contain any header except the mandatory headers shown in the table below:

***Table 1: Mandatory Headers List***

| Mandtory Headers List | | |
|---|---|---|
| ALSO | AUTHORIZATION | CALLID |
| CC_DIVERSION | CC_REDIRECT | CONTACT |
| CONTENT_DISP | CONTENT_ENCODING | CONTENT_LENGTH |
| CONTENT_TYPE | CISCO_GCID | CISCO_GUID |
| CSEQ | DATE | FROM |
| MAX_FORWARDS | MIME_VER | MIME_VER_VAL |
| PRIVACY | PRIVACY_ASSERTED_ID | PRIVACY_PREFERRED_ID |
| PROXY_AUTH | PROXY_AUTHENTICATE | RECORD_ROUTE |
| ROUTE | RTP_STAT | SESSION_EXPIRES |
| TIMESTAMP | TO | USER_AGENT |
| VIA | WWW_AUTHENTICATE | |

### Example

The following example shows how to configure a pass-through list using the **voice class sip-hdr-passthrulist** command and add the header name 'Resource-priority' to the list using the **passthru-hdr** command:

```
Device> enable
Device# configure terminal
Device(config)# voice class sip-hdr-passthrulist 101
Device(config-class)# passthru-hdr Resource-Priority
Device(config-class)# end
```

## Related Commands

| Command | Description |
|---|---|
| **pass-thru** | Passes the Session Description Protocol (SDP) transparently from in-leg to the out-leg with no media negotiation. |
| **passthru-hdr-unsupp** | Enables the pass-thru of all unsupported headers. |
| **voice class sip-hdr-passthrulist** | Configures list of headers to be passed through. |
| **voice-classsip pass-thru** | Passes the Session Description Protocol (SDP) transparently from in-leg to the out-leg with no media negotiation. |

# passthru-hdr-unsupp

To add the unsupported headers to a configured pass-through list and enable the pass-thru of all unsupported headers in the list, use the **passthru-hdr-unsupp** command in voice class configuration mode. To remove the unsupported headers from a configured pass-through list, use the **no** form of the command.

**passthru-hdr-unsupp**
**no passthru-hdr-unsupp**

## Syntax Description

This command has no arguments or keywords.

## Command Default

Unsupported headers are not included in the configured pass-through list.

## Command Modes

Voice class configuration mode (config-class)

## Command History

| Release | Modification |
|---|---|
| 15.4(1)T | This command was introduced. |
| Cisco IOS XE Bengaluru 17.4.1a | Introduced support for YANG models. |

## Usage Guidelines

A pass-through list using the **voice class sip-hdr-passthrulist** command must be configured before adding the unsupported headers to the list.

### Example

The following example shows how to configure a pass-through list using the **voice class sip-hdr-passthrulist** command and add the unsupported headers to the list using the **passthru-hdr-unsupp** command:

```
Device> enable
Device# configure terminal
Device(config)# voice class sip-hdr-passthrulist 100
Device(config-class)# passthru-hdr-unsupp
Device(config-class)# end
```

## Related Commands

| Command | Description |
|---|---|
| **pass-thru** | Passes the Session Description Protocol (SDP) transparently from in-leg to the out-leg with no media negotiation. |
| **passthru-hdr** | Adds a header name to a configured pass-through list. |
| **voice class sip-hdr-passthrulist** | Configures list of headers to be passed through. |
| **voice-classsip pass-thru** | Passes the Session Description Protocol (SDP) transparently from in-leg to the out-leg with no media negotiation. |

# pattern

To match a call based on the entire Session Initiation Protocol (SIP) or telephone (TEL) uniform resource identifier (URI), use the **pattern**command in voice URI class configuration mode. To remove the match, use the **no** form of this command.

**pattern** *uri-pattern*
**no pattern**

## Syntax Description

| | |
|---|---|
| *uri-pattern* | Cisco IOS regular expression (regex) pattern that matches the entire URI. Can be up to 128 characters. |

## Command Default

No default behavior or values

## Command Modes

Voice URI class configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(4)T | This command was introduced. |

## Usage Guidelines

- This command matches a regular expression pattern to the entire URI.
- When you use this command in a URI voice class, you cannot use any other pattern-matching command such as the **host**, **phone context**, **phone number**, or **user-id** commands.

## Examples

The following example configures the voice class to match the entire SIP URI:

```
voice class uri r100 sip
 pattern elmo@cisco.com
```

## Related Commands

| Command | Description |
|---|---|
| **destination uri** | Specifies the voice class to use for matching the destination URI that is supplied by a voice application. |
| **host** | Matches a call based on the host field in a SIP URI. |
| **incoming uri** | Specifies the voice class used to match a VoIP dial peer to the URI of an incoming call. |
| **phone context** | Filters out URIs that do not contain a phone-context field that matches the configured pattern. |
| **phone number** | Matches a call based on the phone number field in a TEL URI. |
| **show dialplan incall uri** | Displays which dial peer is matched for a specific URI in an incoming voice call. |

| Command | Description |
|---|---|
| show dialplan uri | Displays which outbound dial peer is matched for a specific destination URI. |
| **user-id** | Matches a call based on the user-id field in the SIP URI. |
| **voice class uri** | Creates or modifies a voice class for matching dial peers to calls containing a SIP or TEL URI. |