



Cisco Unified Survivable Remote Site Telephony 8.5 New Features

Last Updated: November 05, 2010

This document describes the following new and enhanced features in Cisco Unified Survivable Remote Site Telephony 8.5 (Cisco Unified SRST):

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Cisco Unified SRST 8.5” section on page 74](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Cisco Unified SRST 8.5, page 2](#)
- [Information About Cisco Unified SRST 8.5, page 2](#)
- [How to Configure Cisco Unified SRST 8.5 New Features, page 49](#)
- [Additional References, page 55](#)



Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2009 Cisco Systems, Inc. All rights reserved.

- [Command Reference, page 57](#)
- [Feature Information for Cisco Unified SRST 8.5, page 74](#)

Prerequisites for Cisco Unified SRST 8.5

- Cisco Unified SRST 8.5
- Cisco IOS Release 15.1(3)T

Information About Cisco Unified SRST 8.5

To configure Cisco Unified SRST features, you should understand the following concepts:

- [E.164 Enhancements, page 2](#)
- [Enhancement to Voice Hunt Group Restriction, page 3](#)
- [Forced Authorization Code, page 3](#)
- [Overlap Dialing Support for SCCP IP Phones, page 9](#)
- [XML API for Cisco Unified SRST, page 9](#)

E.164 Enhancements

Cisco Unified SRST 8.5 allows you to present a phone number in + E.164 telephone numbering format. E.164 is an International Telecommunication Union (ITU-T) recommendation that defines the international public telecommunication numbering plan used in the PSTN and other data networks. E.164 defines the format of telephone numbers. A leading + E.164 telephone number can have a maximum of 15 digits and is usually written with a '+' prefix defining the international access code. To dial such numbers from a normal fixed line phone, the appropriate international call prefix must be used.

The leading +E.164 number is unique number specified to a phone or a device. Callers from around the world dial the leading + E.164 phone number to reach a phone or a device without the need to know local or international prefix. The leading + E.164 feature also reduces the overall telephony configuration process by eliminating the need to further translate the telephone numbers.

Phone Registration with Leading + E164 Number

In Cisco Unified SRST, phones register using the leading '+' dialing plan in two ways. Phones can either register with the extension number or with leading + E.164 number.

When phones are registered with extension number, the phones will have a dial peer association with the extension number. The **dialplan-pattern** command is enhanced to allow you to configure leading + phone numbers on the dialplan pattern. Once dialplan-pattern is configured, there could be an E.164 number dialpeer associated with the same phone.

For example, phones registered with extension number 1111 can also be reached by dialing +13332221111. This phone registration method is beneficial in two ways, that is, locally, phones are able to reach each other by just dialing the extension numbers and, remotely, phones can dial abbreviated numbers which are translated as an E.164 number at the outgoing dial-peer.

When phones are registered with a leading + E.164 number, there is only one leading + E.164 number associated with the phone. The **demote** option in the **dialplan-pattern** command allows the phone to have two dialpeers associated with the same phone. For more information on configuring the dialplan-patterns, see, [How to Configure Dialing Plans](#).

Enhancement to Voice Hunt Group Restriction

In Cisco Unified CME 8.5 and later versions, when **call forward no answer (CFNA)** command is configured in a voice hunt group, you are not required to enter a timeout value for voice hunt group member and the **call forward no answer** timer is ignored. In earlier version of Cisco Unified SRST, if **call forward no answer** was configured for a voice hunt group member, you were required to set the value of **timeout** command. The **timeout** value was required to be less than the **timeout** value of the **call-forward no answer** command.

Forced Authorization Code

Forced Authorization Code Overview

Cisco Unified SRST 8.5 allow you to manage call access and call accounting through the Forced Authorization Code (FAC) feature. The FAC feature regulates the type of call a certain caller may place and forces the caller to enter a valid authorization code on the phone before the call is placed. FAC allows you to track callers dialing non-toll-free numbers, long distance numbers, and also for accounting and billing purposes.

In Cisco Unified SRST and Cisco Voice Gateways, devices and endpoints are logically partitioned into different logical partitioning class of restriction (LPCOR) groups. For example, IP phones, Analog phones, PSTN trunks, and IP (h323/SIP) trunks as shown in [Figure 1 on page 4](#), are partitioned into five LPCOR groups under the voice lpcor custom mode, such as:

- voice lpcor custom
 - group 10 Manager
 - group 11 LocalUser
 - group 12 RemoteUser
 - group 13 PSTNTrunk
 - group 14 IPTrunk

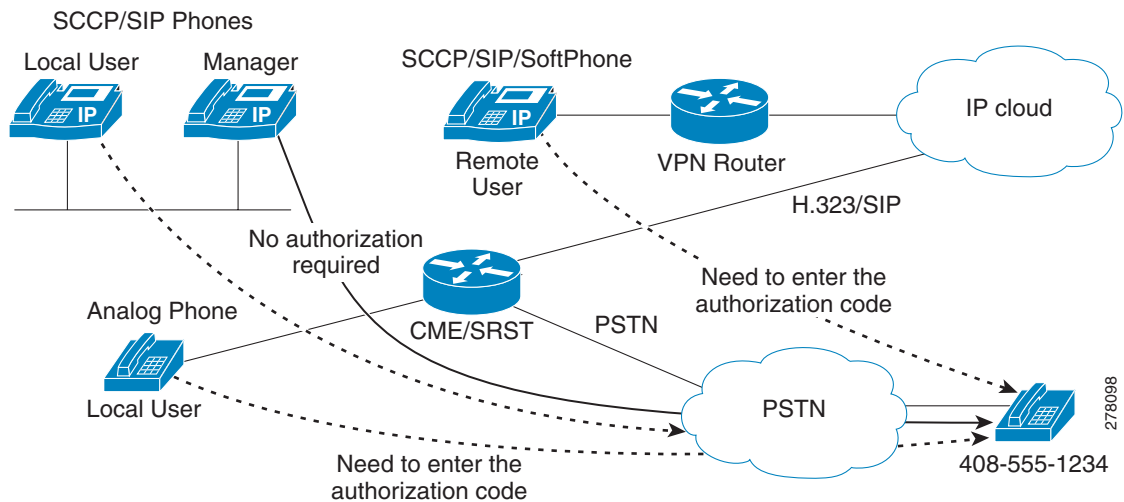


Figure 1 Forced Authorization Code Network Overview

For each group, the LPCOR group policy of a routing endpoint is enhanced to define incoming calls from individual LPCOR groups that are restricted by FAC. A LPCOR group call to a destination is accepted only when a valid FAC is entered. FAC service for a routing endpoint is enabled through the **service fac** defined in a LPCOR group policy. For more information, see [Enabling Forced Authorization Code \(FAC\) on LPCOR Groups](#), page 49.

The following are the group policy rules applicable to the PSTNTrunk LPCOR group:

- FAC is required by PSTNTrunk if a call is initiated from either LocalUser or RemoteUser group.
- Any calls from Manager group are allowed to terminate to PSTNTrunk without restriction.
- Any incoming calls from either IPTrunk or PSTNTrunk group are rejected and terminated to PSTNTrunk group.

For information on configuring LPCOR groups and associating LPCOR group with different device types, see [Call Restriction Regulations](#).

FAC Call Flow

FAC is required for an incoming call based on the LPCOR policy defined for the call destination. Once the authentication is finished, the success or failure status and the collected FAC digits are saved to the call detail records (CDRs).

Calls are handled by a new built-in application authorization package which first plays a user-prompt for the caller to enter a username (in digits) then, the application plays a passwd-prompt for the caller to collect the password (in digits). The collected username and password digits are then used for FAC, see [Defining Parameters for Authorization Package](#), page 51.

When FAC authentication is successful, the outgoing call setup is continued to the same destination. If FAC authentication fails, the call is then forwarded to the next destination. FAC operations are invoked to the call if FAC service is enabled in the next destination and no valid FAC status is saved for the call.

Any calls failing because of FAC blocking are disconnected with a LPCOR Q.850 disconnect cause code. Once the FAC is invoked for a call, the collected authorization digits and the authentication status information is collected by call active or call history records. You can retrieve the FAC information through the **show call active voice** and **show call history voice** commands.

Forced Authorization Code Specification

The authorization code used for call authentication must follow these specifications:

- The authorization code must be in numeric (0 – 9) format.
- A digit collection operation must be completed if either one of the following conditions occur:
 - maximum number of digits are collected
 - digit input times out
 - a terminating digit is entered

Once digit collection is completed, the authentication is done by either the external Radius server or Cisco Unified SRST or Cisco Voice Gateways by using AAA Login Authentication setup. For more information on AAA login authentication methods, see [Configuring Login Authentication Using AAA](#).

When authentication is done by local Cisco Unified CME, Cisco Unified SRST, or Cisco Voice Gateways, the **username ac-code password 0 password** command is required to authenticate the collected authorization code digits.

FAC data is stored through the CDR and new AAA **fac-digits** and **fac-status** attributes and are supported in a CDR STOP record. This CDR STOP record is formatted for file accounting, RADIUS or Syslog accounting purpose.

FAC Requirement for Different Types of Calls

Table 1 shows FAC support for different types of calls.

Table 1 *Fac Support for different types of calls*

Types of Calls	FAC Behavior for Different Calls
Basic Call	A calls B. B requires A to enter a FAC. A is routed to B only when A enters a valid FAC.
Call Forward All Call Forward Busy	When A (with no FAC) calls B, A is call forwarded to C: <ul style="list-style-type: none"> • No FAC is required when B enables Call Forward All or Call Forward Busy to C. • FAC is required on A when A is call forwarded to C.
Call Forward No Answer	When A (with no FAC) calls B and A (with FAC) calls C: A calls B: <ul style="list-style-type: none"> • No FAC is required when A calls B. A is Call Forward No Answer (CFNA) to C. <ul style="list-style-type: none"> • FAC is required on A when A is call forward to C.
Call Transfer (Blind)	FAC is required, if B calls C and A , and A calls C. Example: A calls B. B answers the call. B initiates a blind transfer call to C. A is prompted to enter FAC. A is routed to C only if a valid FAC is entered by A.

Table 1 **Fac Support for different types of calls**

Types of Calls	FAC Behavior for Different Calls
Call Transfer (Consultation) Transfer Complete at Alerting State	<ol style="list-style-type: none"> 1. FAC is required if B calls C. FAC is not required when A calls C. <p>Example:</p> <ol style="list-style-type: none"> a. A calls B. B answers the call and initiates a consultation transfer to C. b. B is prompted to enter a FAC and B is not allowed to complete the call transfer when FAC is not completed. c. B (the transfer call) is forwarded to C after a valid FAC is entered. B completes the transfer while the transfer call is still ringing on C. A is then transferred to C. <ol style="list-style-type: none"> 2. FAC is required if B calls C and A calls C. <p>Example:</p> <ol style="list-style-type: none"> a. A calls B. B answers the call and initiates a consultation transfer to C. b. B is prompted to enter a FAC and B is not allowed to complete the call transfer when FAC is not completed. c. No FAC is required to A, A is then transferred to C. <ol style="list-style-type: none"> 3. FAC is not required if B calls C but FAC is required if A calls C. <p>Example:</p> <ol style="list-style-type: none"> a. A calls B, B answers the call. b. B initiates a consultation transfer to C and completes the transfer. c. No FAC required to A, A is then transferred to C.
Transfer Complete at Connected State	<ol style="list-style-type: none"> 1. FAC is required when A calls C. <p>Example:</p> <ol style="list-style-type: none"> a. A calls B, B answers the call and initiates a consultation transfer to C. b. C answers the transfer call and B completes the transfer. c. No FAC required to connect to A (including local hairpin calls because the call transfer is complete) and A is connected to C.

Table 1 Fac Support for different types of calls

Types of Calls	FAC Behavior for Different Calls
Conference Call (Software/Adhoc)	<ol style="list-style-type: none"> 1. FAC is not invoked when a call is joined to a conference connection. 2. FAC is required between A and C, B and C. <p>Example:</p> <ol style="list-style-type: none"> a. A calls B, B answers the call and initiates a conference call to C. b. B enters a valid authorization code and is routed to C. c. C answers the conference call and the conference is complete. d. No FAC is required to connect to A and A is joined to a conference connection.
Meetme Conference	<ol style="list-style-type: none"> 1. FAC is not invoked for a caller to join the meetme conference. 2. FAC is required between A and C, B and C. <p>Example:</p> <ol style="list-style-type: none"> a. C joins the meetme conference first. b. No FAC is required if B joins the same meetme conference. c. No FAC is required if C also joins the same meetme conference.
Call Park and Retrieval	<ol style="list-style-type: none"> 1. FAC is not invoked for the parked call. 2. FAC is required if C calls A. <p>Example:</p> <ol style="list-style-type: none"> a. A calls B, B answers the call and parks the caller on A. b. C retrieves the parked call (A), no FAC is required to reach C, and C is connected to A.
Call Park Restore	<ol style="list-style-type: none"> 1. FAC is required if A calls D. <p>Example:</p> <ol style="list-style-type: none"> a. A calls B, B answers the call and parks the caller on A. b. Parked call (A) is timed out from a call-park slot and is forwarded to D. c. No FAC is required for D and the parked call (A) will ring on D.

Table 1 *Fac Support for different types of calls*

Types of Calls	FAC Behavior for Different Calls
Group Pickup	<ol style="list-style-type: none"> 1. FAC is not provided if a caller picks up a group call. 2. FAC is required if C calls A. <p>Example:</p> <ol style="list-style-type: none"> a. A calls B, A is ringing on B, and C attempts to pickup call A. b. No FAC is required for C and C is connected to A.
Single Number Redirection (SNR)	FAC is not supported for an SNR call.
Third Party Call Control (3pcc)	FAC is not supported for a three-party call control (3pcc) outgoing call.
Parallel Hunt Groups	FAC is not supported on parallel hunt groups.
Whisper intercom	FAC is not supported for whisper intercom calls.

Overlap Dialing Support for SCCP IP Phones

Cisco Unified SRST 8.5 and later versions support overlap dialing on SCCP IP phones such as, 7942, 7945, 7962, 7965, 7970, 7971, and 7975.

In earlier version of Cisco Unified SRST, overlap dialing was not supported over PRI/BRI trunks for calls originating from SCCP IP phones. The dialing always converted to enbloc dialing based on the dial-peer configuration and the dial-peer mapping application. Once the dialpeer matching takes place, no further dialing was possible and no overlap digit were sent over ISDN trunk, even though overlap dialing was supported over ISDN trunks.

SCCP IP phones currently support overlap dialing, but digits are converted to enbloc digits when it reaches the system. Overlap dialing is supported on SIP IP phones using the KeyPad Markup Language (KPML) method.

In Cisco IOS, with the overlap dialing support, the dialed digits from the SIP or SCCP IP phones is passed across to the PRI/BRI trunks as overlap digits and not as enbloc digits. thus, enabling overlap dialing on the PRI/BRI trunks as well.

You can configure overlap dialing on SCCP IP phones in Cisco Unified SRST. For more information, see, [“Configuring Overlap Dialing on SCCP IP Phones in Cisco Unified SRST”](#) section on page 54.

XML API for Cisco Unified SRST

Cisco Unified SRST 8.5 and later versions adds support for the eXtensible Markup Language (XML) Application Programming Interface (API). This feature has the following sections:

- [Target Audience, page 10](#)
- [Prerequisites, page 10](#)
- [Information About the XML API, page 10](#)
- [Examples, page 13](#)

Target Audience

This document assumes that you have knowledge of a high-level programming language, such as C++, Java, or an equivalent language. You must also have knowledge or experience in the following areas:

- TCP/IP Protocol
- Hypertext Transport Protocol
- Socket programming
- XML

In addition, users of this programming guide must have a firm grasp of XML Schema, which is used to define the AXL requests, responses, and errors. For more information on XML Schema, please see the *XML Schema Part 0: Primer Second Edition*.

Prerequisites

- For Cisco Unified CME: XML API must be configured in Cisco Unified CME. For configuration information, see the “[Configuring the XML API](#)” section of the *Cisco Unified CME Administrator Guide*.

Information About the XML API

The XML API support in Cisco Unified CME and Cisco Unified SRST provides a mechanism for inserting, retrieving, updating, and removing data from the Cisco router using eXtensible Markup Language (XML).

Request methods are XML structures that are passed to the XML server in Cisco Unified CME and Cisco Unified SRST using HTTP POST. The XML server receives the XML structures and executes the request. If the request completes successfully, then the appropriate XML response is returned.

[Table 2](#) lists the request and response methods for the XML API along with the purpose and printers for each method.

Table 2 XML API Methods: Request and Response

Description	Request	Parameter	Response
System			
Execute configuration commands	ISexecCLI	<i>command</i>	ISexecCLIResult
Save router configuration to nvram	ISSaveConfig	—	ISSaveConfigResult
SCCP			
Get system status for Cisco Unified CME or Cisco Unified SRST.	ISgetGlobal	—	ISGlobal

Description	Request	Parameter	Response
Get status of an IP phone	ISgetDevice	Any combination of the following: ISDevID ISDevName ISKeyword: <ul style="list-style-type: none"> - all - allTag - available 	ISDevices
Get configuration of phone template	ISgetDeviceTemplate	Any combination of the following: ISDevTemplateID ISKeyword: <ul style="list-style-type: none"> - all - allTag - available 	ISDeviceTemplates
Get configuration Get configuration of an extension	ISgetExtension	Any combination of the following: ISExtID ISExtNumber ISKeyword: <ul style="list-style-type: none"> - all - allTag - available 	ISExtensions
Get configuration of an extension template	ISgetExtensionTemplate	Any combination of the following: ISExtTemplateID ISKeyword: <ul style="list-style-type: none"> - all - allTag - available 	ISExtensionTemplates
Get user information	ISgetUser	ISuserID	ISuser
Get user profile information	ISgetUserProfile	Any combination of the following: ISUserProfileID ISuserID ISKeyword: <ul style="list-style-type: none"> - all - allTag - available 	ISuserProfiles

Description	Request	Parameter	Response
Get configuration for utility directory	ISgetUtilityDirectory	ISgetUtilityDirectory —	ISUtilityDirectory
SIP			
Get system status for a Cisco Unified CME running SIP	ISgetVoiceRegGlobal	—	ISSipGlobal
Get status of an IP phone	ISgetSipDevice	Any combination of the following: ISPoolID ISPoolName ISKeyword: <ul style="list-style-type: none"> - all - allTag - available 	ISSipDevices
Get configuration of an extension	ISgetSipExtension	Any combination of the following: ISVoiceRegDNID ISVoiceRegNumber ISKeyword: <ul style="list-style-type: none"> - all - allTag - available 	ISSipExtensions
Get status of a session server	ISgetSessionServer	Any combination of the following: ISSessionServerID ISSessionServerName ISKeyword: <ul style="list-style-type: none"> - all - allTag - available 	ISSessionServers
Get status of voice hunt groups	ISgetVoiceHuntGroup	ISVoiceHuntGroupID ISKeyword: <ul style="list-style-type: none"> - all - allTag - available 	ISVoiceHuntGroups
Get configuration for Presence	ISgetPresenceGlobal	—	ISPresenceGlobal

Examples

This section contains the following examples for the following XML API methods:

System

- [ISexecCLI](#)
- [ISSaveConfig](#)

SCCP IP Phones

- [ISgetGlobal](#)
- [ISgetDevice](#)
- [ISgetDeviceTemplate](#)
- [ISgetExtension](#)
- [ISgetExtensionTemplate](#)
- [ISgetUser](#)
- [ISgetUserProfile](#)
- [ISgetUtilityDirectory](#)

SIP IP Phones

- [ISgetVoiceRegGlobal](#)
- [ISgetSipDevice](#)
- [ISgetSipExtension](#)
- [ISgetSessionServer](#)
- [ISgetVoiceHuntGroup](#)
- [ISgetPresenceGlobal](#)

ISexecCLI

Use ISexecCLI to execute a list of Cisco IOS commands on the Cisco router. The request must include the CLI parameter with the Cisco IOS command string for each command to be executed.

Request: Example

```
<SOAP-ENV:Envelope>
<SOAP-ENV:Body>
<axl>
<request xsi:type="ISexecCLI">
<ISexecCLI>
<CLI>ephone 4</CLI>
<CLI>mac-address 00D.BC80.EB51</CLI>
<CLI>type 7960</CLI>
<CLI>button 1:1</CLI>
</ISexecCLI>
</request>
</axl>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response: Example

The value of “0” for ISexecCLIResponse in the following example shows the response when the request is completed successfully.

```
<SOAP-ENV:Envelope >
<SOAP-ENV:Body>
<axl >
<response xsi:type="ISexecCLIResponse" >
<ISexecCLIResponse>0</ISexecCLIResponse>
<ISexecCLIError></ISexecCLIError>
</response>
</axl>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The following example shows the response when the request fails. The value of ISexecCLIResponse identifies which line number in the request failed. Any subsequent commands in the list of commands are not executed. All preceding commands in the list were executed.

```
<SOAP-ENV:Envelope >
<SOAP-ENV:Body>
<axl >
<response xsi:type="ISexecCLIResponse" >
<ISexecCLIResponse>4</ISexecCLIResponse>
<ISexecCLIError> invalid input dn parameter for button 1</ISexecCLIError>
</response>
</axl>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

ISSaveConfig

Use ISSaveConfig to save the running configuration on a router to the startup configuration on the same router.

Request: Example

```
<request>
<ISSaveConfig />
</request>
```

Response: Example

The following example shows that the ISSaveConfig request was successfully completed.

```
<response xsi:type=" ISSaveConfig">
<ISSaveConfigResult>success</ISSaveConfigResult>
</request>
```

The following example shows the response when the request fails.

```
<response xsi:type=" ISSaveConfig">
<ISSaveConfigResult>fail</ISSaveConfigResult>
</request>
```

The following example shows that response when the request is delayed, typically because there is another terminal session connected to Cisco Unified CME. The running configuration will be saved later by a background process after all other terminal sessions are disconnected.

```
<response xsi:type="ISSaveConfig">
<ISSaveConfigResult>delay</ISSaveConfigResult>
</request>
```

ISgetGlobal

Use ISgetGlobal to retrieve system configuration and status information for the Cisco Unified SRST system.

Request: Example

```
<request xsi:type="ISgetGlobal">
<ISgetGlobal></ISgetGlobal>
</request>
```

Response: Example

```

<response>
<ISGlobal>
<ISAddress>10.4.188.90</ISAddress>
<ISMode>ITS</ISMode>
<ISVersion>7.2</ISVersion>
<ISDeviceRegistered>0</ISDeviceRegistered>
<ISPeakDeviceRegistered>1</ISPeakDeviceRegistered>
<ISPeakDeviceRegisteredTime>9470</ISPeakDeviceRegisteredTime>
<ISKeepAliveInterval>30</ISKeepAliveInterval>
<ISConfiguredDevice>32</ISConfiguredDevice>
<ISConfiguredExtension>74</ISConfiguredExtension>
<ISServiceEngine>0.0.0.0</ISServiceEngine>
<ISName>ngm-2800</ISName>
<ISPortNumber>2000</ISPortNumber>
<ISMaxConference>8</ISMaxConference>
<ISMaxRedirect>10</ISMaxRedirect>
<ISMaxEphone>48</ISMaxEphone>
<ISMaxDN>180</ISMaxDN>
<ISVoiceMail>6050</ISVoiceMail>
<ISUrlServices>
<ISUrlService>
<ISUrlType>EPHONE_URL_INFO</ISUrlType>
<ISUrlLink>http://1.4.188.101/localdir</ISUrlLink>
</ISUrlService>
<ISUrlService>
<ISUrlType>EPHONE_URL_DIRECTORIES</ISUrlType>
<ISUrlLink>http://1.4.188.101/localdir</ISUrlLink>
</ISUrlService>
<ISUrlService>
<ISUrlType>EPHONE_URL_MESSAGES</ISUrlType>
<ISUrlLink>http://1.4.188.101/localdir</ISUrlLink>
</ISUrlService>
<ISUrlService>
<ISUrlType>EPHONE_URL_SERVICES</ISUrlType>
<ISUrlLink>http://1.4.188.101/localdir</ISUrlLink>
</ISUrlService>
<ISUrlService>
<ISUrlType>EPHONE_URL_PROXYSERV</ISUrlType>
<ISUrlLink>http://1.4.188.101/localdir</ISUrlLink>
</ISUrlService>
<ISUrlService>
<ISUrlType>EPHONE_URL_IDLE</ISUrlType>
<ISUrlLink>http://1.4.188.101/localdir</ISUrlLink>
</ISUrlService>
<ISUrlService>
<ISUrlType>EPHONE_URL_AUTH</ISUrlType>
<ISUrlLink>http://1.4.188.101/localdir</ISUrlLink>
</ISUrlService>
</ISUrlServices>
<global-after-hours>
<block_list>
<block_item>
<pattern_id>1</pattern_id>
<blocking_pattern>1234</blocking_pattern>
<blocking_option />
</block_item>
<block_item>
<pattern_id>2</pattern_id>
<blocking_pattern>2345</blocking_pattern>
<blocking_option>7-24</blocking_option>
</block_item>

```



```

</block_list>
<date_list>
<date_item>
<month>Nov</month>
<day_of_month>12</day_of_month>
<start_time>12:00</start_time>
<stop_time>13:00</stop_time>
</date_item>
</date_list>
<day_list>
<day_item>
<day_of_week>Mon</day_of_week>
<start_time>12:00</start_time>
<stop_time>13:00</stop_time>
</day_item>
</day_list>
<after-hours_login>
<http>>true</http>
</after-hours_login>
<override-code>2222</override-code>
<pstn-prefix_list>
<pstn-prefix_item>
<index>1</index>
<pstn-prefix>22</pstn-prefix>
</pstn-prefix_item>
</pstn-prefix_list>
</global-after-hours>
<application_name>calling</application_name>
<auth_credential_list>
<credential_item>
<index>1</index>
<user>test</user>
<password>test</password>
</credential_item>
</auth_credential_list>
<auto>
<assign_list>
<assign_item>
<group_id>1</group_id>
<start_tag>70</start_tag>
<stop_tag>93</stop_tag>
<type>anl</type>
<cfw />
<timeout>0</timeout>
</assign_item>
<assign_item>
<group_id>2</group_id>
<start_tag>1</start_tag>
<stop_tag>20</stop_tag>
<cfw>1234</cfw>
<timeout>80</timeout>
</assign_item>
</assign_list>
</auto>
<auto-reg-ephone>>true</auto-reg-ephone>
<bulk-speed-dial_list>
<bulk-speed-dial_item>
<list>1</list>
<url />
</bulk-speed-dial_item>
</bulk-speed-dial_list>
<prefix>123</prefix>
<global-call-forward>
<pattern_list>

```

```

<pattern_item>
<index>2</index>
<pattern>.T</pattern>
</pattern_item>
</pattern_list>
<callfwd_system>
<redirecting-expanded>>false</redirecting-expanded>
</callfwd_system>
</global-call-forward>
<call-park>
<select>
<no-auto-match>>true</no-auto-match>
</select>
<application_system>>true</application_system>
<redirect_system>>true</redirect_system>
</call-park>
<caller-id>
<block_code>*1</block_code>
<name-only>>true</name-only>
</caller-id>
<calling-number>
<initiator>>true</initiator>
<local>>false</local>
<secondary>>false</secondary>
</calling-number>
<cnf-file>
<location>
<TFTP>flash:/its/</TFTP>
<flash>>true</flash>
</location>
<option>perphonetype</option>
</cnf-file>
<default_codec>Unknown</default_codec>
<conference>
<hardware>>true</hardware>
</conference>
<date-format>mm-dd-yy</date-format>
<device-security-mode>none</device-security-mode>
<dialplan-pattern_list>
<dialplan-pattern_item>
<index>1</index>
<pattern>1234</pattern>
<extension-length>4</extension-length>
<extension-pattern />
<demote>>false</demote>
<no-reg>>false</no-reg>
</dialplan-pattern_item>
<dialplan-pattern_item>
<index>2</index>
<pattern>1233</pattern>
<extension-length>4</extension-length>
<extension-pattern />
<demote>>true</demote>
<no-reg>>false</no-reg>
</dialplan-pattern_item>
<dialplan-pattern_item>
<index>3</index>
<pattern>1232</pattern>
<extension-length>4</extension-length>
<extension-pattern>1111</extension-pattern>
<demote>>false</demote>
<no-reg>>false</no-reg>
</dialplan-pattern_item>
<dialplan-pattern_item>

```

```

<index>4</index>
<pattern>1231</pattern>
<extension-length>4</extension-length>
<extension-pattern />
<demote>>false</demote>
<no-reg>>true</no-reg>
</dialplan-pattern_item>
</dialplan-pattern_list>
<directory>
<entry_list>
<entry_item>
<tag>1</tag>
<number>1234</number>
<name>directory</name>
</entry_item>
</entry_list>
<option>last-name-first</option>
</directory>
<dn-webedit>>false</dn-webedit>
<em>
<external>>true</external>
<keep-history>>true</keep-history>
<logout>12:00 00:-1 -1:-1</logout>
</em>
<ephone-reg>>true</ephone-reg>
<extension-assigner>
<tag-type>provision-tag</tag-type>
</extension-assigner>
<fac>
<standard>>true</standard>
<custom_list>
<custom_item>
<fac_string>callfwd all</fac_string>
<fac_list>**1</fac_list>
<alias>0</alias>
<alias_map />
</custom_item>
<custom_item>
<fac_string>callfwd cancel</fac_string>
<fac_list>**2</fac_list>
<alias>0</alias>
<alias_map />
</custom_item>
<custom_item>
<fac_string>pickup local</fac_string>
<fac_list>**3</fac_list>
<alias>0</alias>
<alias_map />
</custom_item>
<custom_item>
<fac_string>pickup group</fac_string>
<fac_list>**4</fac_list>
<alias>0</alias>
<alias_map />
</custom_item>
<custom_item>
<fac_string>pickup direct</fac_string>
<fac_list>**5</fac_list>
<alias>0</alias>
<alias_map />
</custom_item>
<custom_item>
<fac_string>park</fac_string>
<fac_list>**6</fac_list>

```

```

<alias>0</alias>
<alias_map />
</custom_item>
<custom_item>
<fac_string>dnd</fac_string>
<fac_list>**7</fac_list>
<alias>0</alias>
<alias_map />
</custom_item>
<custom_item>
<fac_string>redial</fac_string>
<fac_list>**8</fac_list>
<alias>0</alias>
<alias_map />
</custom_item>
<custom_item>
<fac_string>voicemail</fac_string>
<fac_list>**9</fac_list>
<alias>0</alias>
<alias_map />
</custom_item>
<custom_item>
<fac_string>ephone-hunt join</fac_string>
<fac_list>*3</fac_list>
<alias>0</alias>
<alias_map />
</custom_item>
<custom_item>
<fac_string>ephone-hunt cancel</fac_string>
<fac_list>#3</fac_list>
<alias>0</alias>
<alias_map />
</custom_item>
<custom_item>
<fac_string>ephone-hunt hlog</fac_string>
<fac_list>*4</fac_list>
<alias>0</alias>
<alias_map />
</custom_item>
<custom_item>
<fac_string>ephone-hunt hlog-phone</fac_string>
<fac_list>*5</fac_list>
<alias>0</alias>
<alias_map />
</custom_item>
<custom_item>
<fac_string>trnsfvm</fac_string>
<fac_list>*6</fac_list>
<alias>0</alias>
<alias_map />
</custom_item>
<custom_item>
<fac_string>dpark-retrieval</fac_string>
<fac_list>*0</fac_list>
<alias>0</alias>
<alias_map />
</custom_item>
<custom_item>
<fac_string>cancel call waiting</fac_string>
<fac_list>*1</fac_list>
<alias>0</alias>
<alias_map />
</custom_item>
</custom_list>

```

```

</fac>
<fxo>
<hook-flash>>true</hook-flash>
</fxo>
<hunt-group>
<logout>HLog</logout>
<report>
<url_info>
<prefix>tftp://223.255.254.253/ngm/huntgp/2800/data</prefix>
<hg_suffix>
<low>-1</low>
<high>0</high>
</hg_suffix>
</url_info>
<delay>0</delay>
<duration>24</duration>
<internal>
<duration>5</duration>
<hg_suffix>
<low>1</low>
<high>5</high>
</hg_suffix>
</internal>
</report>
</hunt-group>
<internal-call>
<moh-group>-1</moh-group>
</internal-call>
<ip>
<qos>
<dscp_list>
<dscp_item>
<index>0</index>
<af11>media</af11>
</dscp_item>
<dscp_item>
<index>1</index>
<af12>signal</af12>
</dscp_item>
<dscp_item>
<index>2</index>
<af13>video</af13>
</dscp_item>
<dscp_item>
<index>3</index>
<af21>service</af21>
</dscp_item>
<dscp_item>
<index>4</index>
<af22>media</af22>
</dscp_item>
<dscp_item>
<index>5</index>
<af23>media</af23>
</dscp_item>
<dscp_item>
<index>6</index>
<af31>media</af31>
</dscp_item>
<dscp_item>
<index>7</index>
<af32>media</af32>
</dscp_item>
<dscp_item>

```

```

<index>8</index>
<af33>media</af33>
</dscp_item>
<dscp_item>
<index>9</index>
<af41>media</af41>
</dscp_item>
<dscp_item>
<index>10</index>
<af42>media</af42>
</dscp_item>
<dscp_item>
<index>11</index>
<af43>media</af43>
</dscp_item>
<dscp_item>
<index>12</index>
<cs1>media</cs1>
</dscp_item>
<dscp_item>
<index>13</index>
<cs2>media</cs2>
</dscp_item>
<dscp_item>
<index>14</index>
<cs3>media</cs3>
</dscp_item>
<dscp_item>
<index>15</index>
<cs4>media</cs4>
</dscp_item>
<dscp_item>
<index>16</index>
<cs5>media</cs5>
</dscp_item>
<dscp_item>
<index>17</index>
<cs6>media</cs6>
</dscp_item>
<dscp_item>
<index>18</index>
<cs7>media</cs7>
</dscp_item>
<dscp_item>
<index>19</index>
<default>media</default>
</dscp_item>
<dscp_item>
<index>20</index>
<ef>media</ef>
</dscp_item>
</dscp_list>
</qos>
<source-address>
<primary>10.4.188.90</primary>
<port>2000</port>
<secondary>1.4.188.90</secondary>
<rehome>0</rehome>
<strict-match>true</strict-match>
</source-address>
</ip>
<keepalive>
<timeout>30</timeout>
<aux_timeout>30</aux_timeout>

```

```

</keepalive>
<live-record>999</live-record>
<load_list>
<phone_7914>hehe</phone_7914>
<phone_7915-12>hehe</phone_7915-12>
<phone_7915-24>hehe</phone_7915-24>
<phone_7916-12>hehe</phone_7916-12>
<phone_7916-24>hehe</phone_7916-24>
<phone_12SP>hehe</phone_12SP>
<phone_7902>hehe</phone_7902>
<phone_7906>hehe</phone_7906>
<phone_7910>hehe</phone_7910>
<phone_7911>SCCP11.9-0-1FT6-4DEV</phone_7911>
<phone_7912>hehe</phone_7912>
<phone_7920>hehe</phone_7920>
<phone_7921>hehe</phone_7921>
<phone_7925>hehe</phone_7925>
<phone_7931>hehe</phone_7931>
<phone_7935>hehe</phone_7935>
<phone_7936>hehe</phone_7936>
<phone_7937>hehe</phone_7937>
<phone_7960-7940>P00308000501</phone_7960-7940>
<phone_7941>hehe</phone_7941>
<phone_7941GE>hehe</phone_7941GE>
<phone_7942>hehe</phone_7942>
<phone_7961>SCCP41.8-4-2-38S</phone_7961>
<phone_7962>hehe</phone_7962>
<phone_7965>hehe</phone_7965>
<phone_7970>hehe</phone_7970>
<phone_7971>hehe</phone_7971>
<phone_7975>hehe</phone_7975>
<phone_7985>hehe</phone_7985>
<phone_ata>hehe</phone_ata>
<phone_6921>hehe</phone_6921>
<phone_6941>hehe</phone_6941>
<phone_6961>hehe</phone_6961>
</load_list>
<load-cfg-file_list>
<load-cfg-file_item>
<cfg_file>flash:its/vrf1/XMLDefaultCIPC.cnf.xml</cfg_file>
<alias>cnf.xml</alias>
<sign>>false</sign>
</load-cfg-file_item>
</load-cfg-file_list>
<log>
<table>
<max-size>150</max-size>
<retain-timer>15</retain-timer>
</table>
</log>
<login>
<timeout>60</timeout>
<clear>24:0</clear>
</login>
<max-conferences>
<count>8</count>
<gain>-6</gain>
</max-conferences>
<max-dn>
<count>180</count>
<global_preference>0</global_preference>
<no-reg>secondary</no-reg>
</max-dn>
<max-ephones>48</max-ephones>

```

```

<max-redirect>10</max-redirect>
<modem>
<passthrough>
<payload-type>100</payload-type>
</passthrough>
<relay_sse>
<payload-type>118</payload-type>
</relay_sse>
<relay_sprt>
<payload-type>120</payload-type>
</relay_sprt>
</modem>
<moh_file>flash:music-on-hold.au</moh_file>
<moh-file-buffer>10000</moh-file-buffer>
<multicast>
<moh_ipaddr>239.10.10.10</moh_ipaddr>
<port>2000</port>
<route_list>
<route_item>
<index>1</index>
<route>10.10.10.10</route>
</route_item>
</route_list>
</multicast>
<mwi-server>
<prefix />
<reg-e164>>true</reg-e164>
<relay>>true</relay>
</mwi-server>
<network-locale_list>
<network-locale_item>
<index>0</index>
<locale>US</locale>
</network-locale_item>
<network-locale_item>
<index>1</index>
<locale>US</locale>
</network-locale_item>
<network-locale_item>
<index>2</index>
<locale>US</locale>
</network-locale_item>
<network-locale_item>
<index>3</index>
<locale>US</locale>
</network-locale_item>
<network-locale_item>
<index>4</index>
<locale>US</locale>
</network-locale_item>
</network-locale_list>
<night-service>
<option>everyday</option>
<code>*234</code>
<date_list>
<date_item>
<index>1</index>
<month>Jan</month>
<day_of_month>1</day_of_month>
<start_time>12:00</start_time>
<stop_time>14:00</stop_time>
</date_item>
</date_list>
<day_list>

```



```

<day_item>
<index>1</index>
<day_of_week>Sun</day_of_week>
<start_time>12:00</start_time>
<stop_time>16:00</stop_time>
</day_item>
<day_item>
<index>2</index>
<day_of_week>Mon</day_of_week>
<start_time>12:00</start_time>
<stop_time>16:00</stop_time>
</day_item>
<day_item>
<index>3</index>
<day_of_week>Tue</day_of_week>
<start_time>12:00</start_time>
<stop_time>16:00</stop_time>
</day_item>
<day_item>
<index>4</index>
<day_of_week>Wed</day_of_week>
<start_time>12:00</start_time>
<stop_time>16:00</stop_time>
</day_item>
<day_item>
<index>5</index>
<day_of_week>Thu</day_of_week>
<start_time>12:00</start_time>
<stop_time>16:00</stop_time>
</day_item>
<day_item>
<index>6</index>
<day_of_week>Fri</day_of_week>
<start_time>12:00</start_time>
<stop_time>16:00</stop_time>
</day_item>
<day_item>
<index>7</index>
<day_of_week>Sat</day_of_week>
<start_time>12:00</start_time>
<stop_time>16:00</stop_time>
</day_item>
</day_list>
<everyday>
<start_time>12:00</start_time>
<stop_time>16:00</stop_time>
</everyday>
<weekday>
<start_time>12:00</start_time>
<stop_time>16:00</stop_time>
</weekday>
<weekend>
<start_time>12:00</start_time>
<stop_time>16:00</stop_time>
</weekend>
</night-service>
<pin>1234</pin>
<pin_override>true</pin_override>
<privacy>true</privacy>
<privacy-on-hold>>false</privacy-on-hold>
<protocol>
<mode>dual-stack</mode>
<preference>ipv4</preference>
</protocol>

```

```

<sdspfarm>
  <conference_options>
    <mute-on>124</mute-on>
    <mute-off>234</mute-off>
    <hardware>>false</hardware>
  </conference_options>
  <units>4</units>
  <tag_list>
    <tag_item>
      <tag>1</tag>
      <device>mtp-conf</device>
    </tag_item>
  </tag_list>
  <transcode>
    <sessions>4</sessions>
  </transcode>
  <unregister>
    <force>1</force>
  </unregister>
</sdspfarm>
<secondary-dialtone>4567</secondary-dialtone>
<secure-signaling>
  <trustpoint />
</secure-signaling>
<server-security-mode />
<service>
  <local-directory>>true</local-directory>
  <local-directory_authenticate>>false</local-directory_authenticate>
  <dss>>false</dss>
  <dnis>
    <overlay>>false</overlay>
    <dir-lookup>>false</dir-lookup>
  </dnis>
  <directed-pickup>>true</directed-pickup>
  <directed-pickup_gpickup>>false</directed-pickup_gpickup>
  <phone_list>
    <phone_item>
      <index>1</index>
      <phone_params>displayOnTime</phone_params>
      <phone_text>time.xml</phone_text>
    </phone_item>
  </phone_list>
</service>
  <ssh>
    <userid>ngm</userid>
    <password>ngm</password>
  </ssh>
  <standby>
    <user>ngm</user>
    <password>ngm</password>
  </standby>
  <system_message>LITTLE TWIN STARS (2800)</system_message>
  <tftp-server-credentials>
    <trustpoint />
  </tftp-server-credentials>
  <time-format>12</time-format>
  <time-webedit>>false</time-webedit>
  <time-zone>0</time-zone>
  <timeouts>
    <busy_timeout>10</busy_timeout>
    <interdigit_timeout>10</interdigit_timeout>
    <ringing_timeout>180</ringing_timeout>
    <transfer-recall_timeout>0</transfer-recall_timeout>
    <night-service-bell_timeout>12</night-service-bell_timeout>

```

```

</timeouts>
<transfer-digit-collect>new-call</transfer-digit-collect>
<transfer-pattern_list>
<transfer-pattern_item>
<index>1</index>
<pattern>...</pattern>
<blind>>false</blind>
</transfer-pattern_item>
<transfer-pattern_item>
<index>2</index>
<pattern>.T</pattern>
<blind>>false</blind>
</transfer-pattern_item>
</transfer-pattern_list>
<transfer-system>
<type>full-consult</type>
<dss>>false</dss>
</transfer-system>
<trunk_optimization_pre_connect>>false</trunk_optimization_pre_connect>
<url_list>
<information>
<url>http://1.4.188.101/localdir</url>
</information>
<directories>
<url>http://1.4.188.101/localdir</url>
</directories>
<messages>
<url>http://1.4.188.101/localdir</url>
</messages>
<services>
<url>http://1.4.188.101/localdir</url>
<name />
</services>
<proxy_server>
<url>http://1.4.188.101/localdir</url>
</proxy_server>
<idle>
<url>http://1.4.188.101/localdir</url>
<idle_timeout>90</idle_timeout>
</idle>
<authentication>
<url>http://1.4.188.101/localdir</url>
<user />
<password />
</authentication>
</url_list>
<user-locale_list>
<user-locale_item>
<index>0</index>
<locale>US</locale>
<package>en</package>
<load />
</user-locale_item>
<user-locale_item>
<index>1</index>
<locale>US</locale>
<package>en</package>
<load />
</user-locale_item>
<user-locale_item>
<index>2</index>
<locale>US</locale>
<package>en</package>
<load />

```

```

</user-locale_item>
<user-locale_item>
<index>3</index>
<locale>US</locale>
<package>en</package>
<load />
</user-locale_item>
<user-locale_item>
<index>4</index>
<locale>US</locale>
<package>en</package>
<load />
</user-locale_item>
</user-locale_list>
<video>
<maximum>
<bit-rate>10000000</bit-rate>
</maximum>
</video>
<voicemail>6050</voicemail>
<web>
<system_admin>
<name>Admin</name>
<secret>-1</secret>
<password />
</system_admin>
<customer_admin>
<name>ngm</name>
<secret>5</secret>
<password>$1$.nfD$zn3h3bp/4grULFS87ZHHV</password>
</customer_admin>
<customize>
<load />
</customize>
</web>
<xml>
<user>cisco</user>
<password>cisco</password>
<level>0</level>
</xml>
</ISGlobal>
</response>

```

ISgetDevice

Use ISgetDevice to retrieve configuration and status information for IP phones.

Use any combination of the following parameters in the request message to specific one or more SCCP phones:

- ISDevID with the ephone tag number of SCCP phone to be queried.
- ISDevName with the MAC address of SCCP phone to be queried.
- ISKeyword with one of the following options:
 - all—All configured SCCP phones
 - allTag— Ephone tag numbers for all SCCP phones configured
 - available— Next available ephone tag number to be configured

Request: Example

```

<request xsi:type="ISgetDevice">
<ISgetDevice>
<ISDevID>1</ISDevID>
<ISDevName>SEP0012DA8AC43D</ISDevName>
<ISDevName>allKeyphone</ISDevName>
</ISgetDevice>
</request>

```

Response: Example

```

<response>
<ISDevices>
<ISDevice>
<ISDevID>1</ISDevID>
<ISDevName>SEP0016C7C7AF9D</ISDevName>
<ISDevType>Others</ISDevType>
<ISconfigDevType>7911</ISconfigDevType>
<ISDevUsername>test</ISDevUsername>
<ISDevLineButtons>
<ISDevLineButton>
<ISDevLineButtonID>1</ISDevLineButtonID>
<ISDevLineButtonMode>MONITOR_RING</ISDevLineButtonMode>
</ISDevLineButton>
</ISDevLineButtons>
<after-hours_exempt>>false</after-hours_exempt>
<after-hours_login>
<http>>false</http>
</after-hours_login>
<block-blind-xf-fallback>>false</block-blind-xf-fallback>
<capf-ip-in-cnf>>false</capf-ip-in-cnf>
<codec>
<codec_name>g711ulaw</codec_name>
<dspfarm-assist>>false</dspfarm-assist>
</codec>
<adhoc_conference>
<add-mode>
<creator>>true</creator>
</add-mode>
<admin>>true</admin>
<drop-mode>
<creator>>false</creator>
<local>>false</local>
</drop-mode>
</adhoc_conference>
<fastdial_list>
<fastdial_item>
<fastdial>1</fastdial>
<fastdial_number>1234</fastdial_number>
<fastdial_name>home LINE</fastdial_name>
</fastdial_item>
</fastdial_list>
<feature-button_list>
<feature-button_item>
<feature-button>1</feature-button>
<feature_type>Dnd</feature_type>
</feature-button_item>
<feature-button_item>
<feature-button>2</feature-button>
<feature_type>Flash</feature_type>
</feature-button_item>
</feature-button_list>

```

```

<keep-conference>
<hangup>true</hangup>
<drop-last>>false</drop-last>
<endcall>true</endcall>
<local-only>true</local-only>
</keep-conference>
<keypad-normalize>>false</keypad-normalize>
<keyphone>>false</keyphone>
<mtp>true</mtp>
<multicast-moh>true</multicast-moh>
<night-service_bell>true</night-service_bell>
<privacy />
<privacy-button>>false</privacy-button>
<transfer-park>
<blocked>>false</blocked>
</transfer-park>
<transfer-pattern>
<blocked>>false</blocked>
</transfer-pattern>
<busy-trigger-per-button>0</busy-trigger-per-button>
<emergency-resp_location>0</emergency-resp_location>
<max-calls-per-button>0</max-calls-per-button>
<nte-end-digit-delay>0</nte-end-digit-delay>
<keepalive>
<timeout>30</timeout>
<aux_timeout>30</aux_timeout>
</keepalive>
<lpcor>
<type>none</type>
</lpcor>
<exclude-services>
<em_service>true</em_service>
<directory_service>>false</directory_service>
<myphoneapp_service>>false</myphoneapp_service>
</exclude-services>
<park>
<reservation-group>park</reservation-group>
</park>
<paging-dn>
<dn>0</dn>
<mode>multicast</mode>
</paging-dn>
<speed-dial_list>
<speed-dial_item>
<index>1</index>
<phone_number>1234</phone_number>
<label>home</label>
</speed-dial_item>
</speed-dial_list>
<ssh>
<userid>ngm</userid>
<password>ngm</password>
</ssh>
<phone_type>
<name>7911</name>
<addon_list>
<addon_item>
<addon>1</addon>
<addon_type>7914</addon_type>
</addon_item>
</addon_list>
</phone_type>
<auto-line>
<mode>normal</mode>

```

```

<auto_select_line>0</auto_select_line>
</auto-line>
<blf-speed-dial_list>
<blf-speed-dial_item>
<index>1</index>
<phone_number>1234</phone_number>
<label>blfsd</label>
</blf-speed-dial_item>
<device>>true</device>
</blf-speed-dial_list>
<bulk-speed-dial_list>
<bulk-speed-dial_item>
<list>1</list>
<url />
</bulk-speed-dial_item>
</bulk-speed-dial_list>
<capf-auth-str>7777</capf-auth-str>
<description>ephoneOne</description>
<device-security-mode>none</device-security-mode>
<dnd>
<feature-ring>>true</feature-ring>
</dnd>
<ephone-template>1</ephone-template>
<headset>
<auto-answer>
<line_list>
<line>1</line>
</line_list>
</auto-answer>
</headset>
<logout-profile>0</logout-profile>
<display_all_missed_calls>true</display_all_missed_calls>
<mwi-line>1</mwi-line>
<offhook-guard-timer>0</offhook-guard-timer>
<phone-ui>
<snr>true</snr>
<speeddial-fastdial>true</speeddial-fastdial>
</phone-ui>
<pin>1234</pin>
<presence>
<call-list>true</call-list>
</presence>
<provision-tag>1</provision-tag>
<username>test</username>
<password>test</password>
<video_enable>true</video_enable>
<vm-device-id>SEP0016C7C7AF9D</vm-device-id>
<ISDevAddr>
<Xipv4Address>0.0.0.0</Xipv4Address>
</ISDevAddr>
<ISPhoneLineList>
<ExtMapStatus>
<LineId>1</LineId>
<ExtId>176</ExtId>
<ExtNumber>6176</ExtNumber>
<ExtStatus>>false</ExtStatus>
<LineState>idle</LineState>
</ExtMapStatus>
</ISPhoneLineList>
<ISKeyPhone>>false</ISKeyPhone>
<SNRui>true</SNRui>
<ISLogoutProfileID>0</ISLogoutProfileID>
<ISUserProfileID>0</ISUserProfileID>
<ISTapiClientAddr>

```

```

<Xipv4Address />
</ISTapiClientAddr>
<ISDevStatus>unregistered</ISDevStatus>
<ISDevLastStatus>deceased</ISDevLastStatus>
<ISDevChangeTime>4040</ISDevChangeTime>
<ISDevKeepAlives>0</ISDevKeepAlives>
<ISDevTapiCStatus />
<ISTapiCLastStatus />
<ISTapiCChangeTime />
<ISTapiCKeepAlive />
<ISDevDND>no</ISDevDND>
</ISDevice>
</ISDevices>
</response>

```

ISgetDeviceTemplate

Use ISgetDeviceTemplate to retrieve configuration and status information for IP phone templates. Use any combination of the following parameters in the request message to specify one or more phone templates:

- ISDevTemplateID with phone template tag number to be queried
- ISKeyword with one of the following options:
 - all— All configured phone templates
 - allTag— Phone template tag numbers for all configured phone templates
 - available—Next available phone template tag number to be configured

Request: Example

```

<request>
<ISgetDeviceTemplate>
<ISgetDevTemplateID>1</ISgetDevTemplateID>
<ISgetDeviceTemplate>
</request>

```


Response: Example

```

<response>
<ISDeviceTemplates>
<ISDeviceTemplate>
<ISDevTemplateID>1</ISDevTemplateID>
<after-hours>
<block_list>
<block_item>
<pattern_id>1</pattern_id>
<blocking_pattern>1234</blocking_pattern>
<blocking_option>7-24</blocking_option>
</block_item>
</block_list>
<date_list>
<date_item>
<month>Jan</month>
<day_of_month>1</day_of_month>
<start_time>12:00</start_time>
<stop_time>14:00</stop_time>
</date_item>
</date_list>
<day_list>
<day_item>
<day_of_week>Mon</day_of_week>
<start_time>12:00</start_time>
<stop_time>14:00</stop_time>
</day_item>
</day_list>
<exempt>>true</exempt>
<after-hours_login>
<http>>true</http>
</after-hours_login>
<override-code>1234</override-code>
</after-hours>
<block-blind-xf-fallback>>false</block-blind-xf-fallback>
<button-layout_phone_7931>0</button-layout_phone_7931>
<button-layout_list>
<button-layout_item>
<button-layout>1,9</button-layout>
<button-type>line</button-type>
</button-layout_item>
<button-layout_item>
<button-layout>4-5,7</button-layout>
<button-type>speed-dial</button-type>
</button-layout_item>
<button-layout_item>
<button-layout>2-3</button-layout>
<button-type>feature</button-type>
</button-layout_item>
<button-layout_item>
<button-layout>11</button-layout>
<button-type>url</button-type>
</button-layout_item>
</button-layout_list>
<capf-ip-in-cnfg>>false</capf-ip-in-cnfg>
<codec>
<codec_name>g711ulaw</codec_name>
<dspfarm-assist>>false</dspfarm-assist>
</codec>
<adhoc_conference>
<add-mode>
<creator>>false</creator>

```

```

</add-mode>
<admin>>false</admin>
<drop-mode>
<creator>>false</creator>
<local>>false</local>
</drop-mode>
</adhoc_conference>
<fastdial_list>
<fastdial_item>
<fastdial>1</fastdial>
<fastdial_number>1234</fastdial_number>
<fastdial_name>office</fastdial_name>
</fastdial_item>
</fastdial_list>
<feature-button_list>
<feature-button_item>
<feature-button>1</feature-button>
<feature_type>HLog</feature_type>
</feature-button_item>
<feature-button_item>
<feature-button>2</feature-button>
<feature_type>Park</feature_type>
</feature-button_item>
<feature-button_item>
<feature-button>3</feature-button>
<feature_type>Privacy</feature_type>
</feature-button_item>
</feature-button_list>
<url-button_list>
<url-button_item>
<url-button>1</url-button>
<url-button_type>em</url-button_type>
</url-button_item>
<url-button_item>
<url-button>3</url-button>
<url-button_type>myphoneapp</url-button_type>
</url-button_item>
<url-button_item>
<url-button>6</url-button>
<url-button_type>service</url-button_type>
<url-button_url>hello</url-button_url>
<url-button_name>helloworld</url-button_name>
</url-button_item>
</url-button_list>
<features_blocked>Pickup Park GPickup</features_blocked>
<keep-conference>
<hangup>>false</hangup>
<drop-last>>false</drop-last>
<endcall>>false</endcall>
<local-only>>false</local-only>
</keep-conference>
<keypad-normalize>>false</keypad-normalize>
<keyphone>>false</keyphone>
<mlpp>
<indication>>true</indication>
<preemption>>true</preemption>
<max_priority>-1</max_priority>
</mlpp>
<mtp>>false</mtp>
<multicast-moh>>true</multicast-moh>
<night-service_bell>>false</night-service_bell>
<privacy />
<privacy-button>>false</privacy-button>
<phone_service>

```

```

<param_list>
<param_item>
<param>displayOnTime</param>
<text>170</text>
</param_item>
</param_list>
</phone_service>
<softkeys>
<alerting_keys />
<connected_keys />
<hold_keys />
<idle_keys />
<remote-in-use_keys>CBarge Newcall</remote-in-use_keys>
<ringing_keys />
<seized_keys />
</softkeys>
<transfer-park>
<blocked>>false</blocked>
</transfer-park>
<transfer-pattern>
<blocked>>false</blocked>
</transfer-pattern>
<busy-trigger-per-button>0</busy-trigger-per-button>
<emergency-resp_location>0</emergency-resp_location>
<max-calls-per-button>0</max-calls-per-button>
<network_locale>0</network_locale>
<nte-end-digit-delay>0</nte-end-digit-delay>
<transfer_max-length>0</transfer_max-length>
<user_locale>0</user_locale>
<keepalive>
<timeout>30</timeout>
<aux_timeout>30</aux_timeout>
</keepalive>
<lpcor>
<type>none</type>
</lpcor>
<exclude-services>
<em_service>>false</em_service>
<directory_service>>true</directory_service>
<myphoneapp_service>>true</myphoneapp_service>
</exclude-services>
<park>
<reservation-group>1234</reservation-group>
</park>
<paging-dn>
<dn>0</dn>
<mode>multicast</mode>
</paging-dn>
<speed-dial_list>
<speed-dial_item>
<index>1</index>
<phone_number>1234</phone_number>
<label>play</label>
</speed-dial_item>
</speed-dial_list>
<ssh>
<userid>test</userid>
<password>test</password>
</ssh>
<phone_type>
<name>7960</name>
<addon_list>
<addon_item>
<addon>1</addon>

```

```

<addon_type>7914</addon_type>
</addon_item>
</addon_list>
</phone_type>
<url_services_list>
<url_services_item>
<services_id>1</services_id>
<url>http</url>
<name>HTTP</name>
</url_services_item>
</url_services_list>
</ISDeviceTemplate>
</ISDeviceTemplates>
</response>

```

ISgetExtension

Use ISgetExtension to retrieve configuration and status information for extension numbers.

Use any combination of the following parameters in the request message to specify one or more extensions:

- ISExtID with the extension ID number to be queried.
- ISExtNumber with the extension number to be queried.
- ISKeyword with one of the following options:
 - all—Displays details of all extension numbers configured
 - allTag— Displays a list of all extension ID numbers configured
 - available— Next available extension ID number to be configured

Request: Example

```

<request>
<ISExtension>
<ISVExtID>1</ISExtID>
<ISExtNumber>1</ISExtNumber>
</ISExtension>
</request>

```

Response: Example

```

<response>
<ISExtensions>
<ISExtension>
<ISExtID>1</ISExtID>
<ISExtNumber>6001</ISExtNumber>
<ISExtSecNumber>6111</ISExtSecNumber>
<ISExtType>normal</ISExtType>
<ISExtStatus>up</ISExtStatus>
<ISExtChangeTime>3122733</ISExtChangeTime>
<ISExtUsage>0</ISExtUsage>
<ISExtHomeAddress>0.0.0.0</ISExtHomeAddress>
<ISExtMultiLines>0</ISExtMultiLines>
<ISExtPortName>EFXS 50/0/1</ISExtPortName>
<ISExtLineMode>DUAL_LINE</ISExtLineMode>
<ISExtCallStatus>IDLE</ISExtCallStatus>
<Mobility>>false</Mobility>
<SNRnumber>1111</SNRnumber>

```

```

<SNRdelay>10</SNRdelay>
<SNRtimeout>5</SNRtimeout>
<SNRnoanNumber />
<ISAllowWatch>true</ISAllowWatch>
<ISSessionServerIDs>
<ISSessionServerID>1</ISSessionServerID>
</ISSessionServerIDs>
<firstName />
<lastName>ephoneDnOne</lastName>
<callForwardAll>1234</callForwardAll>
<ISDevList>
<ISDeviceID>8</ISDeviceID>
</ISDevList>
<allow>
<watch>true</watch>
</allow>
<call-forward>
<all>
<number>1234</number>
</all>
<busy>
<number>9000</number>
<option>secondary</option>
<dialplan-pattern>>false</dialplan-pattern>
</busy>
<max-length>
<number />
</max-length>
<night-service-activated>
<number>2323</number>
</night-service-activated>
<noan>
<number>1234</number>
<timeout>80</timeout>
<dialplan-pattern>true</dialplan-pattern>
<option />
</noan>
</call-forward>
<call-waiting>
<cw_beep>
<accept>true</accept>
<generate>true</generate>
</cw_beep>
<cw_ring>true</cw_ring>
</call-waiting>
<corlist>
<incoming />
<outgoing />
</corlist>
<cti>
<notify>true</notify>
<watch>true</watch>
</cti>
<description>ephoneDnOne</description>
<hold-alert>
<timeout>15</timeout>
<mode>idle</mode>
<ring-silent-dn>true</ring-silent-dn>
</hold-alert>
<huntstop>
<channel>8</channel>
</huntstop>
<moh-group>0</moh-group>
<mwi>

```

```

<type>qsig</type>
<mode />
</mwi>
<mwi-type>both</mwi-type>
<pickup-group />
<transfer-recall_timeout>0</transfer-recall_timeout>
<translate>
<called>1</called>
<calling>2</calling>
</translate>
<translation-profile>
<incoming>in</incoming>
<outgoing>out</outgoing>
</translation-profile>
<application>
<name>calling</name>
<out-bound>calling</out-bound>
</application>
<port-caller-id>
<block>>false</block>
<local>>false</local>
<transfer_passthrough>>false</transfer_passthrough>
</port-caller-id>
<conference_dn>
<mode />
<unlocked>>false</unlocked>
</conference_dn>
<ephone-dn-template>0</ephone-dn-template>
<ephone-hunt_login>>true</ephone-hunt_login>
<feed>
<ip_addr>0.0.0.0</ip_addr>
<port>0</port>
<route>0.0.0.0</route>
<out-call />
</feed>
<fwd-local-calls>>true</fwd-local-calls>
<intercom>
<dn-plar />
<barge-in>>false</barge-in>
<label />
<no-mute>>true</no-mute>
<ptt>>false</ptt>
<no-auto-answer>>true</no-auto-answer>
</intercom>
<label />
<loopback-dn>
<dn>0</dn>
<auto-con>>false</auto-con>
<loopback-codec />
<forward>0</forward>
<prefix />
<retry>0</retry>
<strip>0</strip>
<suffix />
</loopback-dn>
<mailbox-selection>
<last-redirect-num>>false</last-redirect-num>
</mailbox-selection>
<moh>
<ip_addr>0.0.0.0</ip_addr>
<port>0</port>
<route>0.0.0.0</route>
<out-call />
</moh>

```

```

<name>ephoneDnOne</name>
<night-service_bell>>false</night-service_bell>
<telephony_number>
<primary>6001</primary>
<secondary>6111</secondary>
<no-reg>>true</no-reg>
<no-reg_option />
</telephony_number>
<paging>
<group />
<ip_addr>0.0.0.0</ip_addr>
<port>0</port>
</paging>
<park-slot>
<directed>>false</directed>
<reserved-for />
<reservation-group />
<timeout>0</timeout>
<limit>0</limit>
<notify />
<only>>false</only>
<transfer_destination />
<recall>>true</recall>
<alternate />
<retry>0</retry>
<retry_limit>0</retry_limit>
</park-slot>
<pickup-call>
<any-group>>false</any-group>
</pickup-call>
<dn_preference>
<order>0</order>
<secondary>9</secondary>
</dn_preference>
<queueing-dn>
<mode />
<timeout>180</timeout>
<transfer_number />
</queueing-dn>
<ring>
<type>external</type>
<line>primary</line>
</ring>
<session-server>
<server>1</server>
</session-server>
<snr_info>
<value>1111</value>
<delay>10</delay>
<timeout>5</timeout>
<cfwd-noan />
</snr_info>
<transfer-mode />
<trunk>
<number />
<timeout>3</timeout>
<transfer-timeout>0</transfer-timeout>
<monitor-port />
</trunk>
<whisper-intercom>
<speed-dial />
<label />
</whisper-intercom>
</ISExtension>

```

```
</ISExtensions>
</response>
```

ISgetExtensionTemplate

Use the ISgetExtensionTemplates to retrieve configuration and status information for extension templates. Use any combination of the following:

- ISExtTemplateID with the extension template ID number to be queried.
- ISKeyword with one of the following options:
 - all—Displays details of all configured extension templates
 - allTag— Displays a list of all configured extension template ID numbers
 - available— Next available extension template ID number to be configured

Request: Example

```
<request>
<ISExtensionTemplates>
<ISExtensionTemplateID>1</ISExtensionTemplateID>
</ISgetExtensionTemplate>
</request>
```

Response: Example

```
<response>
<ISExtensionTemplates>
<ISExtensionTemplate>
<ISExtTemplateID>1</ISExtTemplateID>
<allow>
<watch>>false</watch>
</allow>
<call-forward>
<all>
<number>1234</number>
</all>
<busy>
<number>3456</number>
<option>primary</option>
<dialplan-pattern>>false</dialplan-pattern>
</busy>
<max-length>
<number>4</number>
</max-length>
<night-service-activated>
<number>7777</number>
</night-service-activated>
<noan>
<number>9999</number>
<timeout>80</timeout>
<dialplan-pattern>>false</dialplan-pattern>
<option>secondary</option>
</noan>
</call-forward>
<call-waiting>
<cw_beep>
<accept>>true</accept>
<generate>>true</generate>
</cw_beep>
```



```

<cw_ring>true</cw_ring>
</call-waiting>
<caller-id_blocked>true</caller-id_blocked>
<corlist>
<incoming />
<outgoing />
</corlist>
<cti>
<notify>>false</notify>
<watch>>false</watch>
</cti>
<description>ephoneDnTemplate</description>
<hold-alert>
<timeout>15</timeout>
<mode>idle</mode>
<ring-silent-dn>true</ring-silent-dn>
</hold-alert>
<huntstop>
<channel>8</channel>
</huntstop>
<moh-group>0</moh-group>
<mwi>
<type>sip</type>
<mode>on-off</mode>
</mwi>
<mwi-type>both</mwi-type>
<pickup-group>1</pickup-group>
<transfer-recall_timeout>400</transfer-recall_timeout>
<translate>
<called>1</called>
<calling>0</calling>
</translate>
<translation-profile>
<incoming>1</incoming>
<outgoing>1</outgoing>
</translation-profile>
</ISExtensionTemplate>
</ISExtensionTemplates>
</response>

```

ISgetUser

Use ISgetUser to retrieve information for a particular user in Cisco Unified CME. The request must include the ISuserID parameter with a user name that is configured in Cisco Unified CME. If the request contains a valid ISuserID, the response includes the user-name tag number (ISuserTag) and type for this user.

The value for ISuserType corresponds to how a username is configured in Cisco Unified CME, as follows:

- 0—INVALID_CME_USER
- 1—EPHONE_USER
- 2—LOGOUT_PROFILE_USER
- 3—USER_PROFILE_USER

If the request contains an invalid ISuserID, the value for ISuserTag and ISuserType will both be “0.”

Request: Example

```
<request>
<ISgetUser>
<ISuserID>a</ISuserID>
</ISgetUser>
</request>
```

Response: Example

```
<response>
<ISuser>
<ISuserID>a</ISuserID>
<ISuserType>3</ISuserType>
<ISuserTag>1</ISuserTag>
</ISuser>
</response>
```

ISgetUserProfile

Use the ISgetUserProfile to retrieve the status and configuration information for a specific user profile. Use any combination of the following:

- ISUserProfileID with the user profile ID of a specific user.
- ISuserID with user ID of a specific user.
- ISKeyword with one of the following options:
 - all—Displays details of all configured user profiles.
 - allTag— Displays a list of all configured user profile IDs.
 - available— Next available user profile.

Request: Example

```
<request>
<ISgetUserProfile>
<ISUserProfileID>1</ISUserProfileID>
</ISgetUserProfile>
</request>
```

Response: Example

```
<response>
<ISUserProfiles>
<ISUserProfile>
<ISUserProfileID>1</ISUserProfileID>
<ISuserID>a</ISuserID>
<ISpassword>a</ISpassword>
<ISuserPin>12</ISuserPin>
<ISPrivacyButton>no</ISPrivacyButton>
<ISuserMaxIdleTime>0</ISuserMaxIdleTime>
<SpeedDials>
<SpeedDial>
<SpeedDialIndex>1</SpeedDialIndex>
<SpeedDialNumber>901</SpeedDialNumber>
<SpeedDialLabel />
<SpeedDialBLF>no</SpeedDialBLF>
</SpeedDial>
<SpeedDial>
```

```

<SpeedDialIndex>2</SpeedDialIndex>
<SpeedDialNumber>902</SpeedDialNumber>
<SpeedDialLabel />
<SpeedDialBLF>no</SpeedDialBLF>
</SpeedDial>
<SpeedDial>
<SpeedDialIndex>3</SpeedDialIndex>
<SpeedDialNumber>2002</SpeedDialNumber>
<SpeedDialLabel>2002Label</SpeedDialLabel>
<SpeedDialBLF>no</SpeedDialBLF>
</SpeedDial>
<SpeedDial>
<SpeedDialIndex>5</SpeedDialIndex>
<SpeedDialNumber>2004</SpeedDialNumber>
<SpeedDialLabel>2004</SpeedDialLabel>
<SpeedDialBLF>yes</SpeedDialBLF>
</SpeedDial>
</SpeedDials>
<UserNumbers>
<UserNumber>
<ISExtNumber>2003</ISExtNumber>
<ISExtMode>NORMAL</ISExtMode>
<ISExtOverlayGroup>0</ISExtOverlayGroup>
<ISExtCombo>no</ISExtCombo>
</UserNumber>
<UserNumber>
<ISExtNumber>201</ISExtNumber>
<ISExtMode>NORMAL</ISExtMode>
<ISExtOverlayGroup>0</ISExtOverlayGroup>
<ISExtCombo>no</ISExtCombo>
</UserNumber>
<UserNumber>
<ISExtNumber>202</ISExtNumber>
<ISExtMode>NORMAL</ISExtMode>
<ISExtOverlayGroup>0</ISExtOverlayGroup>
<ISExtCombo>no</ISExtCombo>
</UserNumber>
</UserNumbers>
<ISuserCurrentPhone>
<CurrentPhoneType>Unknown</CurrentPhoneType>
<CurrentPhoneID>0</CurrentPhoneID>
</ISuserCurrentPhone>
</ISUserProfile>
</ISUserProfiles>
</response>

```

ISgetUtilityDirectory

Use the ISgetUtilityDirectory to retrieve status and configuration information for directory information.

Request: Example

```
<request>
```

```
<ISgetUtilityDirectory>
</ISgetUtilityDirectory>
</request>
```

Response: Example

```
<response>
<ISUtilityDirectory>
<ISDirectoryEntry>
<ISDirectoryTag>1</ISDirectoryTag>
<ISDirectoryNumber>12345</ISDirectoryNumber>
<firstName>first</firstName>
<lastName>last</lastName>
</ISDirectoryEntry>
<ISDirectoryEntry>
<ISDirectoryTag>2</ISDirectoryTag>
<ISDirectoryNumber>67890</ISDirectoryNumber>
<firstName>first2</firstName>
<lastName>last 2</lastName>
</ISDirectoryEntry>
</ISUtilityDirectory>
</response>
```

ISgetVoiceRegGlobal

Use the ISgetVoiceRegGlobal to retrieve status and configuration information of global parameters for SIP,

Request: Example

```
<request>
<ISgetVoiceRegGlobal>
</ISgetVoiceRegGlobal>
</request>
```

Response: Example

```
<response>
<ISSipGlobal>
<ISAddress>10.10.10.1</ISAddress>
<ISMode>cme</ISMode>
<ISVersion>7.1</ISVersion>
<ISAuthModes>
<ISAuthMode>ood_refer</ISAuthMode>
<ISAuthMode>presence</ISAuthMode>
</ISAuthModes>
<ISPortNumber>5060</ISPortNumber>
<ISMaxPool>10</ISMaxPool>
<ISMaxDN>100</ISMaxDN>
<ISMaxRedirect>5</ISMaxRedirect>
</ISSipGlobal>
</response>
```

ISgetSipDevice

For SIP phones, use any combination of the following parameters in the request message to specify one or more SIP phones:

- ISPoolID with the voice register pool tag number of SIP phone to be queried.
- ISPoolName with the voice register pool name of the SIP phone to be queried.
- ISKeyword with one of the following options:
 - all—All configured SIP phones
 - allTag— Voice register pool tag numbers for all configured SIP phones
 - available— Next available phone tag number to be configured

Request: Example

```
<request>
<ISgetSipDevice>
<ISPoolID>1</ISPoolID>
</ISgetSipDevice>
</request>
```

Response: Example

```
<response>
<ISSipDevices>
<ISSipDevice>
<ISPoolID>1</ISPoolID>
<ISDevMac>0013.1978.3CA5</ISDevMac>
<ISSessionServerID>0</ISSessionServerID>
<ISDevAddr>
<Xipv4Address>0</Xipv4Address>
</ISDevAddr>
<ISSipPhoneLineList>
<ExtMapStatus>
<LineId>1</LineId>
<ExtId>1</ExtId>
<ExtNumber>901</ExtNumber>
<LineState>idle</LineState>
</ExtMapStatus>
<ExtMapStatus>
<LineId>2</LineId>
<ExtId>2</ExtId>
<ExtNumber>902</ExtNumber>
<LineState>idle</LineState>
</ExtMapStatus>
</ISSipPhoneLineList>
<ISPoolMaxRegistration>42</ISPoolMaxRegistration>
<ISPoolDtmfRelay>rtp-nte</ISPoolDtmfRelay>
<ISDevCodec>g729r8</ISDevCodec>
</ISSipDevice>
</ISSipDevices>
</response>
```

ISgetSipExtension

Use ISgetSipExtension to retrieve configuration and status information for extension numbers. Use any combination of the following parameters in the request message to specify one or more extensions:

- ISVoiceRegDNID with the extension ID number to be queried
- ISVoiceRegNumber with the extension number to be queried.

- ISKeyword with one of the following options:
 - all—Displays details of all configured extension numbers.
 - allTag— Displays a list of all configured extension numbers.
 - available— Next available extension ID number to be configured.

Request: Example

```
<request>
<ISgetSipExtension>
<ISVoiceRegDNID>1</ISVoiceRegDNID>
</ISgetSipExtension>
</request>
```

Response: Example

```
<response>
<ISSipExtensions>
<ISSipExtension>
<ISVoiceRegDNID>1</ISVoiceRegDNID>
<ISExtNumber>901</ISExtNumber>
<ISSessionServerIDs>
<ISSessionServerID>1</ISSessionServerID>
<ISSessionServerID>2</ISSessionServerID>
</ISSessionServerIDs>
<ISAllowWatch>true</ISAllowWatch>
<firstName>Henry</firstName>
<lastName>Mann</lastName>
<ISSipDevList>
<ISPoolID>1</ISPoolID>
<ISPoolID>2</ISPoolID>
</ISSipDevList>
</ISSipExtension>
</ISSipExtensions>
</response>
```

ISgetSessionServer

Use ISgetSessionServer to retrieve configuration information for session servers in Cisco Unified CME. Use any combination of the following parameters in the request message to specify one or more session servers:

- ISSessionServerID with the session server tag number.
- ISSessionserverName with session server name.
- ISKeyword with one of the following keywords:
 - all—All configured session servers
 - allTag—Session server tag numbers for all configured session servers
 - available—Next available session server tag number to be configured

Request: Example

```
<request>
<ISgetSessionServer>
<ISSessionServerID>1</ISSessionServerID>
</ISgetSessionServer>
</request>
```

Response: Example

```
<response>
<ISSessionServers>
<ISSessionServer>
<ISSessionServerID>1</ISSessionServerID>
<ISSessionRegisterID>SS1</ISSessionRegisterID>
<ISSessionKeepAlives>60</ISSessionKeepAlives>
</ISSessionServer>
</ISSessionServers>
</response>
```

ISgetVoiceHuntGroup

Use the ISgetVoiceHuntGroupID to retrieve status and configuration information for voice hunt groups. Use any combination of the following parameters in the request message to specify one or more voice hunt groups:

- ISVoiceHuntGroupID with the voice hunt group ID number
- ISKeyword with one of the following keywords:
 - all—All configured voice hunt groups
 - allTag—Voice hunt group ID numbers for all configured voice hunt groups
 - available—Next available voice hunt group ID number to be configured

Request: Example

```
<request>
<ISgetVoiceHuntGroup>
<ISVoiceHuntGroupID>1</ISVoiceHuntGroupID>
</ISgetVoiceHuntGroup>
</request>
```

Response: Example

```
<response>
<ISVoiceHuntGroups>
<ISVoiceHuntGroup>
<ISVoiceHuntGroupID>1</ISVoiceHuntGroupID>
<ISVoiceHuntGroupType>longest-idle</ISVoiceHuntGroupType>
<ISVoiceHuntGroupPilotNumber>200</ISVoiceHuntGroupPilotNumber>
<ISVoiceHuntGroupPilotPeerTag>200</ISVoiceHuntGroupPilotPeerTag>
<ISVoiceHuntGroupPilotPreference>0</ISVoiceHuntGroupPilotPreference>
<ISVoiceHuntGroupSecPilotNumber />
<ISVoiceHuntGroupSecPilotPeerTag>-1</ISVoiceHuntGroupSecPilotPeerTag>
<ISVoiceHuntGroupSecPilotPreference>0</ISVoiceHuntGroupSecPilotPreference>
<ISVoiceHuntGroupListSize>2</ISVoiceHuntGroupListSize>
<ISVoiceHuntGroupListNums>
```

```

<ISVoiceHuntGroupListNum>201</ISVoiceHuntGroupListNum>
<ISVoiceHuntGroupListNum>202</ISVoiceHuntGroupListNum>
</ISVoiceHuntGroupListNums>
<ISVoiceHuntGroupFinalNum />
<ISVoiceHuntGroupTimeout>180</ISVoiceHuntGroupTimeout>
<ISVoiceHuntGroupHops>2</ISVoiceHuntGroupHops>
</ISVoiceHuntGroup>
</ISVoiceHuntGroups>
</response>

```

ISgetPresenceGlobal

Use ISgetPresenceGlobal to retrieve configuration information and status for the presence engine in Cisco Unified CME.

Request: Example

```

<request>
<ISgetPresenceGlobal />
</request>

```

Response: Example

```

<response>
<ISPresenceGlobal>
<ISPresenceEnable>true</ISPresenceEnable>
<ISMode>cme</ISMode>
<ISAllowSub>true</ISAllowSub>
<ISAllowWatch>true</ISAllowWatch>
<ISMaxSubAllow>100</ISMaxSubAllow>
<ISSipUaPresenceStatus>>false</ISSipUaPresenceStatus>
</ISPresenceGlobal>
</response>

```


How to Configure Cisco Unified SRST 8.5 New Features

This section contains the following tasks.

- [Enabling Forced Authorization Code \(FAC\) on LPCOR Groups, page 49](#)
- [Defining Parameters for Authorization Package, page 51](#)
- [Configuring Overlap Dialing on SCCP IP Phones in Cisco Unified SRST, page 54](#)

Enabling Forced Authorization Code (FAC) on LPCOR Groups

To enable FAC, perform the following steps.

Prerequisites

- You must enable the **voice lpcor enable** command before configuring FAC.
- Trunks (IP and PSTN) must be associated with phones into different LPCOR groups. See the [Associating a LPCOR Policy with Analog Phone or PSTN Trunk Calls](#) for more information.

Restrictions

- Authenticated FAC data is saved to a call-leg from which the authorization code is collected. When a call-forward or blind transfer call scenario triggers a new call due to the SIP notify feature, the same caller is required to enter the authorization code again for FAC authentication.



A FAC pin code must be unique and not the same as an extension number. Cisco Unified SRST and Cisco Voice Gateways will not validate whether a collected FAC pin code matches an extension number.

SUMMARY STEPS

1. **enables**
2. **configure terminal**
3. **voice lpcor enable**
4. **voice lpcor custom**
5. **group number lpcor-group**
6. **exit**
7. **voice lpcor policy lpcor-group**
8. **accept lpcor-group fac**
9. **service fac**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example: Router> enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example: Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<p>voice lpcor enable</p> <p>Example: Router(config)# voice lpcor enable</p>	<p>Enables LPCOR functionality on the Cisco Unified CME router.</p>
Step 4	<p>voice lpcor custom</p> <p>Example: Router(config)# voice lpcor custom</p>	<p>Defines the name and number of LPCOR resource groups on the Cisco Unified CME router.</p>
Step 5	<p>group number lpcor-group</p> <p>Example: Router(cfg-lpcor-custom)#group 10 Manager Router(cfg-lpcor-custom)#group 11 LocalUser Router(cfg-lpcor-custom)#group 12 RemoteUser Routercfg-lpcor-custom)#group 13 PSTNTrunk Router(cfg-lpcor-custom)#group 14 IPTrunk</p>	<p>Adds a LPCOR resource group to the custom resource list.</p> <ul style="list-style-type: none"> <i>number</i>—Group number of the LPCOR entry. Range: 1 to 64. <i>lpcor-group</i>—String that identifies the LPCOR resource group.
Step 6	<p>exit</p> <p>Example: Router(conf-voi-serv)# exit</p>	<p>Exits voice-service configuration mode.</p>
Step 7	<p>voice lpcor policy lpcor-group</p> <p>Example: Router(cfg-lpcor-custom)#group 10 Manager Router(cfg-lpcor-custom)#group 11 LocalUser Router(cfg-lpcor-custom)#group 12 RemoteUser Router(cfg-lpcor-custom)#group 13 PSTNTrunk Router(cfg-lpcor-custom)#group 14 IPTrunk</p>	<p>Creates a LPCOR policy for a resource group.</p> <ul style="list-style-type: none"> <i>lpcor-group</i>—Name of the resource group that you defined in Step 5.
Step 8	<p>accept lpcor-group fac</p> <p>Example: Router(cfg-lpcor-policy)# accept PSTNTrunk fac Router(cfg-lpcor-policy)# accept Manager fac</p>	<p>Allows a LPCOR policy to accept calls associated with the specified resource group.</p> <ul style="list-style-type: none"> Default: Calls from other groups are rejected; calls from the same resource group are accepted. <i>fac</i>—Valid forced authorization code that the caller needs to enter before the call is routed to its destination. Repeat this command for each resource group whose calls you want this policy to accept.

	Command or Action	Purpose
Step 9	service <i>fac</i> Example: Router(cfg-lpcor-policy)#service fac	Enables force authorization code service for a LPCOR group. <ul style="list-style-type: none"> • Default: No form of the service fac command is the default setting of a LPCOR group policy.
Step 10	end Example: Router(config-ephone)# end	Returns to privileged EXEC mode.

Examples

```

Router# show voice lpcor policy
voice lpcor policy PSTNTrunk (group 13):
  service fac is enabled
  ( accept      ) Manager (group 10)
  ( reject      ) LocalUser (group 11)
  ( reject      ) RemoteUser (group 12)
  ( accept      ) PSTNTrunk (group 13)
  ( reject      ) IPTrunk (group 14)

```

Defining Parameters for Authorization Package

To define required parameters for user name and password, follow these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **application**
4. **package auth**
5. **param user-prompt** *filename*
6. **param passwd-prompt** *filename*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example: Router> enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example: Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<p>application</p> <p>Example: Router(config)#application Router(config-app)#</p>	<p>Enters the application configuration mode.</p>
Step 4	<p>package auth</p> <p>Example: Router(config-app)#package auth</p>	<p>Enters package authorization configuration mode.</p>
Step 5	<p>param passwd</p> <p>Example: Router(config-app)#package param passwd 12345</p>	<p>Character string that defines a predefined password for authorization.</p> <p>Note Password digits collection is optional if password digits are predefined in the param passwd comamnd</p>
Step 6	<p>param user-prompt filename</p> <p>Example: Router(config-app-param)#param user-prompt flash:en_bacd_enter_dest.au</p>	<p>Allows you to enter the user name parameters required for package authorization for FAC authentication.</p> <ul style="list-style-type: none"> user-prompt filename — Plays an audio prompt requesting the caller to enter a valid username (in digits) for authorization.
Step 7	<p>param passwd-prompt filename</p> <p>Example: Router(config-app-param)#param passwd-prompt flash:en_welcome.au</p>	<p>Allows you to enter the password parameters required for package authorization for FAC authentication.</p> <ul style="list-style-type: none"> passwd-prompt filename— Plays an audio prompt requesting the caller to enter a valid password (in digits) for authorization..
Step 8	<p>param max-entries</p> <p>Example: Router(config-app-param)#param max-entries 0</p>	<p>Specifies number of attempts to re-enter an account or a password</p> <ul style="list-style-type: none"> max-entries—Value ranges from 0-10, default value is 0.
Step 9	<p>param term-digit</p> <p>Example: Router(config-app-param)#param term-digit #</p>	<p>Specifies digit for terminating an account or a password digit collection.</p>

	Command or Action	Purpose
Step 10	param abort-digit Example: Router(config-app-param)#param abort-digit *	Specifies the digit for aborting username or password digit input. Default value is *.
Step 11	param max-digits Example: Router(config-app-param)#param max-digits 32	Maximum number of digits in a username or password. Range of valid value: 1 - 32. Default value is 32.
Step 12	exit Example: Router(conf-app-param)# exit	Exits package authorization parameter configuration mode.

Configuring Overlap Dialing on SCCP IP Phones in Cisco Unified SRST

To configure overlap signaling on SCCP IP phones in Cisco Unified SRST, follow these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **call-manager-fallback**
4. **overlap-signal**
5. **end**

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	call-manager-fallback Example: Router(config) call-manager-fallback	Enters call-manager-fallback configuration mode.
Step 4	overlap-signal Example: Router(config-cm-fallback) #overlap-signal	Allows to configure overlap signaling support for SCCP IP phones.
Step 5	end Example: Router(config-cm-fallbac) # end	Returns to privileged EXEC mode.

Examples:

The following example shows **overlap-signal** configured in call-manager-fallback mode:

```
Router# show run | sec call-manager
call-manager-fallback
max-conferences 12 gain -6
transfer-system full-consult
overlap-signal
```

Additional References

The following sections provide references related to Cisco Unified SRST.

Related Documents

Related Topic	Document Title
Cisco Unified CME configuration	<ul style="list-style-type: none"> <i>Cisco Unified Communications Manager Express System Administrator Guide</i> <i>Cisco Unified Communications Manager Express Command Reference</i>
Cisco Unified CME network design	<ul style="list-style-type: none"> <i>Cisco Unified CallManager Express Solution Reference Network Design Guide</i>
Cisco IOS voice configuration	<ul style="list-style-type: none"> <i>Cisco IOS Voice Configuration Library</i> <i>Cisco IOS Voice Command Reference</i>
Phone documentation for Cisco Unified CME	<ul style="list-style-type: none"> <i>User Documentation for Cisco Unified IP Phones</i>

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	<p>http://www.cisco.com/techsupport</p>

Command Reference

The following commands are introduced or modified in the features documented in this module.

New Commands

- [overlap-signal](#)

Modified Commands

- [dialplan-pattern \(call-manager-fallback\)](#)
- [package](#)
- [param](#)

overlap-signal

To configure overlap dialing in SCCP or SIP IP phones, use the **overlap-signal** command in ephone, ephone-template, telephony-service, voice register pool, voice register global, or voice register template configuration mode.

overlap-signal

Syntax Description This command has no arguments or keywords.

Command Default Overlap-signal is disabled.

Command Modes

- Call-manager-fallback
- Ephone configuration (config-ephone)
- Ephone-template configuration (config-ephone-template)
- Telephony-service configuration (config-telephony)
- Voice register pool (config-register-pool)
- Voice register global configuration (config-register-global)
- Voice register template (config-register-template)

Command History	Cisco IOS Release	Cisco Product	Modification
	15.1(3)T	Cisco Unified CME 8.5 Cisco Unified SRST 8.5	This command was introduced.

Usage Guidelines

SCCP IP phones

In SCCP IP phones, overlap dialing is enabled when the **overlap signal** command is configured in ephone, ephone-template, and telephony-service configurations modes.

SIP IP phones

In SIP IP Phones, overlap dialing is enabled when the **overlap signal** command is configured in voice register pool, voice register global, and voice register template configuration modes.

Cisco Unified SRST

In Cisco Unified SRST, overlap dialing is enabled on SCCP IP phones when **overlap signal** command is configured in call-manager-fallback configuration mode.

Examples

The following example shows overlap-signal enabled on SCCP phones:

```
Router# show running config
!
!
telephony-service
max-ephones 25
max-dn 15
load 7906 SCCP11.8-5-3S.loads
load 7911 SCCP11.8-5-3S.loads
load 7921 CP7921G-1.3.3.LOADS
load 7941 SCCP41.8-5-3S.loads
load 7942 SCCP42.8-5-3S.loads
load 7961 SCCP41.8-5-3S.loads
load 7962 SCCP42.8-5-3S.loads
max-conferences 12 gain -6
web admin system name cisco password cisco
transfer-system full-consult
create cnf-files version-stamp Jan 01 2002 00:00:00
overlap-signal
!
ephone-template 1
button-layout 1 line
button-layout 3-6 blf-speed-dial
!
ephone-template 9
feature-button 1 Endcall
feature-button 3 Mobility
!
!
ephone-template 10
feature-button 1 Park
feature-button 2 MeetMe
feature-button 3 CallBack
button-layout 1 line
button-layout 2-4 speed-dial
button-layout 5-6 blf-speed-dial
overlap-signal
!
ephone 10
device-security-mode none
mac-address 02EA.EAEA.0010
overlap-signal
!
```

The following example shows **overlap-signal** configured in **voice register global** and **voice register pool 10**:

```
Router#show running config
!
!
!
voice service voip
 ip address trusted list
  ipv4 20.20.20.1
 media flow-around
 allow-connections sip to sip
!
voice class media 10
 media flow-around
!
!
voice register global
 max-pool 10
 overlap-signal
!
voice register pool 5
 overlap-signal
!
!
!
```

The following example shows **overlap-signal** configured in call-manager-fallback mode:

```
Router# show run | sec call-manager
call-manager-fallback
 max-conferences 12 gain -6
 transfer-system full-consult
 overlap-signal
```


dialplan-pattern (call-manager-fallback)

To create a global prefix that can be used to expand the extension numbers of inbound and outbound calls into fully qualified E.164 numbers, use the **dialplan-pattern** command in call-manager-fallback configuration mode. To disable the **dialplan-pattern** command settings, use the **no** form of this command.

dialplan-pattern *tag pattern extension-length extension-length* [**extension-pattern** *extension-pattern*] [**no-reg**] [**demote**]

no dialplan-pattern *tag [pattern extension-length extension-length extension-pattern extension-pattern]* [**no-reg**] [**demote**]

Syntax Description		
	<i>tag</i>	Dial-plan string tag used before a ten-digit telephone number. The tag number is from 1 to 10.
	<i>pattern</i>	Dial-plan pattern, such as the area code, the prefix, and the first one or two digits of the extension number, plus wildcard markers or dots (.) for the remainder of the extension number digits.
	extension-length	Sets the number of extension digits that will appear as a caller ID.
	<i>extension-length</i>	The number of extension digits. The extension length must match the setting for IP phones in Cisco Unified CallManager mode. The range is from 1 to 32.
	extension-pattern	(Optional) Sets an extension number's leading digit pattern when it is different from the E.164 telephone number's leading digits defined in the <i>pattern</i> variable.
	<i>extension-pattern</i>	(Optional) The extension number's leading digit pattern. Consists of one or more digits and wildcard markers or dots (.). For example, 5.. would include extensions 500 to 599; 5... would include extensions 5000 to 5999. The extension pattern must match the setting for IP phones in Cisco Unified CallManager mode.
	no-reg	(Optional) Prevents the E.164 numbers in the dial peer from registering with the gatekeeper.
	demote	(Optional) Demotes the registered phone if it matches the <i>pattern</i> , <i>extension-length</i> , and <i>extension pattern</i> .

Command Default No default behavior or values.

Command Modes Call-manager-fallback configuration

Command History

Cisco IOS Release	Cisco Product	Modification
12.1(5)YD	Cisco SRST 1.0	This command was introduced on the Cisco 2600 series and Cisco 3600 series multiservice routers and on the Cisco IAD2420 series.
12.2(2)XT	Cisco SRST 2.0	This command was implemented on the Cisco 1750 and Cisco 1751 multiservice routers.
12.2(8)T	Cisco SRST 2.0	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on the Cisco 3725 and Cisco 3745 routers.
12.2(8)T1	Cisco SRST 2.0	This command was implemented on the Cisco 2600-XM and Cisco 2691 routers.
12.2(11)T	Cisco SRST 2.01	This command was integrated into Cisco IOS Release 12.2(11)T and implemented on the Cisco 1760 routers.
12.2(11)YT	Cisco SRST 2.1	The extension-pattern keyword was added.
15.1(3)T	Cisco Unified SRST 8.5	This command was modified. The demote keyword was added to the dialplan pattern command and the dialplan pattern tag value was increased to 1-10.

Usage Guidelines

The **dialplan-pattern** command builds additional dial peers. For example, if a hidden POTS dial peer is created, such as the following:

```
Router(config)# dial-peer voice 20001 pots
Router(config-dial-peer)# destination-pattern 1001
Router(config-dial-peer)# voice-port 50/0/2
```

and a dial-plan pattern is created, such as 40855510.., then an additional dial peer will be created that allows calls to both the 1001 and 4085551001 numbers. For example:

```
Router(config)# dial-peer voice 20002 pots
Router(config-dial-peer)# destination-pattern 4085551001
Router(config-dial-peer)# voice-port 50/0/2
```

Both dial peers can be seen with the **show dial-peer voice** command.

The **dialplan-pattern** command also creates a global prefix that can be used by inbound calls (calls to an IP phone in a Cisco Unified SRST system) and outbound calls (calls made from an IP phone in a Cisco Unified SRST system) to expand their extension numbers to fully qualified E.164 numbers.

For inbound calls (calls to an IP phone in a Cisco Unified SRST system) where the calling party number matches the dial-plan pattern, the call is considered a local call and has a distinctive ring that identifies the call as internal. Any calling party number that does not match the dial-plan pattern is considered an external call and has a distinctive ring that is different from the internal ringing.

For outbound calls, the **dialplan-pattern** command converts the calling party's extension number to an E.164 calling party number. Outbound calls that do not use an E.164 number and go through a PRI connection to the PSTN may be rejected by the PRI link as the calling party identifier.

If there are multiple patterns, called-party numbers are checked in numeric order, starting with pattern 1, until a match is found or until the last pattern has been checked. The valid dial-plan pattern with the lowest tag is used as a prefix to all local Cisco IP phones.

When **extension-pattern** *extension-pattern* keyword and argument are used, the leading digits of an extension pattern are stripped and replaced with the corresponding leading digits of the dial plan. For example, the following command maps all extension numbers 4xx to the PSTN number 40855501xx, so that extension 412 corresponds to 4085550112.

```
Router(config)# call-manager-fallback
Router(config-cm-fallback)# dialplan-pattern 1 4085550100 extension-length 3
extension-pattern 4..
```

The number of *extension-pattern* argument characters must match the number set for the *extension-length* argument. For example, if the *extension-length* is 3, the *extension-pattern* can be 8.., 1.., 51., and so forth.

A dial-plan pattern is required to register the Cisco IP phone lines with a gatekeeper. The **no-reg** keyword provides the option of not registering specific numbers to the gatekeeper so that those numbers can be used for other telephony services.

When the **demote** keyword is used, the **dialplan-pattern** command tries to demote the registered phone if it matches the *pattern*, *extension-length*, and *extension-pattern*.

Examples

The following example shows how to create dial-plan pattern 1 for extension numbers 5000 to 5099 with a prefix of 408555. If an inbound calling party number (4085555044) matches dial-plan pattern 1, the recipient phone will display an extension (5044) as the caller ID and use an internal ringing tone. If an outbound calling party extension number (5044) matches dial-plan pattern 1, the calling party extension will be converted to an E.164 number (4085555044). The E.164 calling party number will appear as the caller ID.

```
Router(config)# call-manager-fallback
Router(config-cm-fallback)# dialplan-pattern 1 40855550.. extension-length 4
extension-pattern 50..
```

In the following example, the **dialplan-pattern** command creates dial-plan pattern 1 for extensions 800 to 899 with the telephone prefix starting with 4085559. As each number in the extension pattern is declared with the **number** command, two POTs dial peers are created. In the example, they are 801 (an internal office number) and 4085559001 (an external number).

```
Router(config)# call-manager-fallback
Router(config-cm-fallback)# dialplan-pattern 1 40855590.. extension-length 3
extension-pattern 8..
```

The following example shows a configuration for two Cisco Unified SRST systems. Each is configured with the same **dialplan-pattern** commands, but one system uses 50.. and the other uses 60.. for extension numbers. Calls from the “50..” system to the “60..” system, and vice versa, are treated as internal calls. Calls that go across an H.323 network and calls that go to a PSTN through an ISDN interface on one of the configured Cisco Unified SRST routers are represented as E.164.

```
Router(config)# call-manager-fallback
Router(config-cm-fallback)# dialplan-pattern 1 40855550.. extension-length 4
extension-pattern 50..
Router(config-cm-fallback)# dialplan-pattern 2 51055560.. extension-length 4
extension-pattern 60..
```

Related Commands

Command	Description
call-manager-fallback	Enables Cisco Unified SRST support and enters call-manager-fallback configuration mode.
show dial-peer voice	Displays information for voice dial peers.



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

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

Defaults

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.

Command	Description
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.

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	
<i>param-name</i>	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 <i>filename</i>	(Optional) Announcement URL to request password input. <i>filename</i> defines the name and location of the audio filename to be used for playing the password prompt.
param user-prompt <i>filename</i>	(Optional) Announcement URL to request authorization code username. <i>filename</i> 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.

Defaults 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-discp-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.

Command	Description
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.

Feature Information for Cisco Unified SRST 8.5

Table 3 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 3 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 3 Feature Information for Cisco Unified SRST 8.1

Feature Name	Releases	Feature Information
Cisco Unified SRST 8.5	15.1(3)T	<ul style="list-style-type: none"> • E.164 Enhancements • Enhancement to Voice Hunt Group Restriction • Forced Authorization Code • Overlap Dialing Support for SCCP IP Phones • XML API for Cisco Unified SRST

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2010 Cisco Systems, Inc. All rights reserved.