Polycom Moscow
info@polycom-moscow.ru
T +7-495-924-25-25
www.polycom-moscow.ru

ADMINISTRATOR'S GUIDE 1.1.0 | May 2014 | 3725-68635-001/A

Polycom® CX5500 Unified Conference Station for Microsoft® Lync™

6001 America Center Drive

San Jose, CA 95002

USA

Customer Feedback We are striving to improve our documentation quality and we appreciate your feedback. Email your opinions and comments to DocumentationFeedback@polycom.com.

Visit the Polycom Support Center for End User License Agreements, software downloads, product documents, product licenses, troubleshooting tips, service requests, and more.

Contents

About This Guide

The *Polycom® CX5500 Unified Conference Station Administrator's Guide* provides instructions for installing, provisioning, and administering the CX5500 Unified Conference Station. This guide will help you understand the Polycom VoIP network and telephony components of the CX5500 system, provides descriptions of all available phone features, and helps you perform the following tasks:

- Install and configure your phone on a network server or Web server
- Configure your phone's features and functions
- Configure your phone's user settings
- Troubleshoot common phone issues

Web Info: Using the Polycom CX5500 Unified Conference Station

For more information on how to use the features available on the CX5500 system, see the *Polycom CX5500 Unified Conference Station for Microsoft Lync User Guide*.

Who Should Read This Guide?

System administrators and network engineers should read this guide to learn how properly to set up the CX5500 system. This guide describes administration-level tasks and is not intended for end users.

Before reading this guide, you should be familiar with the following:

- Computer networking and driver administration for your operating system
- An XML editor
- The XML-based configuration file format used for the Polycom UC Software

Conventions Used in This Guide

Polycom guides contains graphical elements and a few typographic conventions. Familiarizing yourself with these elements and conventions will help you successfully perform tasks.

Icons Used in this Guide

Name	*Icon*	*Description*
Note		The Note icon highlights information of interest or important information needed to be successful in accomplishing a procedure or to understand a concept.
Administrator Tip		The Administrator Tip icon highlights techniques, shortcuts, or productivity related tips.

Name	*Icon*	*Description*
Caution		The Caution icon highlights information you need to know to avoid a hazard that could potentially impact device performance, application functionality, or successful feature configuration.
Warning		The Warning icon highlights an action you must perform (or avoid) to prevent issues that may cause you to lose information or your configuration setup, and/or affect phone or network performance.
Web Info		The Web Info icon highlights supplementary information available online such as documents or downloads on support.polycom.com or other locations.
Timesaver		The Timesaver icon highlights a faster or alternative method for accomplishing a method or operation.
Power Tip		The Power Tip icon highlights faster, alternative procedures for advanced administrators already familiar with the techniques being discussed.
Troubleshooting		The Troubleshooting icon highlights information that may help you solve a relevant problem or to refer you to other relevant troubleshooting resources.
Settings		The Settings icon highlights settings you may need to choose for a specific behavior, to enable a specific feature, or to access customization options.

A few typographic conventions, listed next, are used in this guide to distinguish types of in-text information.

Typographic Conventions

Convention	*Description*
Bold	Highlights interface items such as menu selections, soft keys, file names, and directories. Also used to represent menu selections and text entry.
Italics	Used to emphasize text, to show example values or inputs, and to show titles of reference documents available from the Polycom Support web site and other reference sites.
Blue Text	Used for cross references to other sections within this document and for hyperlinks to external documents and web sites.
`Courier`	Used for code fragments and parameter names.

This guide also uses a few writing conventions to distinguish conditional information.

Writing Conventions

Convention	*Description*
<*MACaddress*>	Indicates that you must enter information specific to your installation, phone, or network. For example, when you see <*MACaddress*>, enter your phone's 12-digit MAC address. If you see <*installed-directory*>, enter the path to your installation directory.
>	Indicates that you need to select an item from a menu. For example, **Settings > Basic** indicates that you need to select **Basic** from the **Settings** menu.
`parameter.*`	Used for configuration parameters. If you see a parameter name in the form `parameter.*` , the text is referring to all parameters beginning with `parameter`. See Read the Feature Parameter Tables for an example.

Recommended Software Tools

Polycom recommends that you use an XML editor—such as XML Notepad—to create and edit configuration files. In this way, all configuration files that you create will be valid XML files.

If the configuration files are not valid XML, they will not load on the handset and an error message will be logged to the provisioning server.

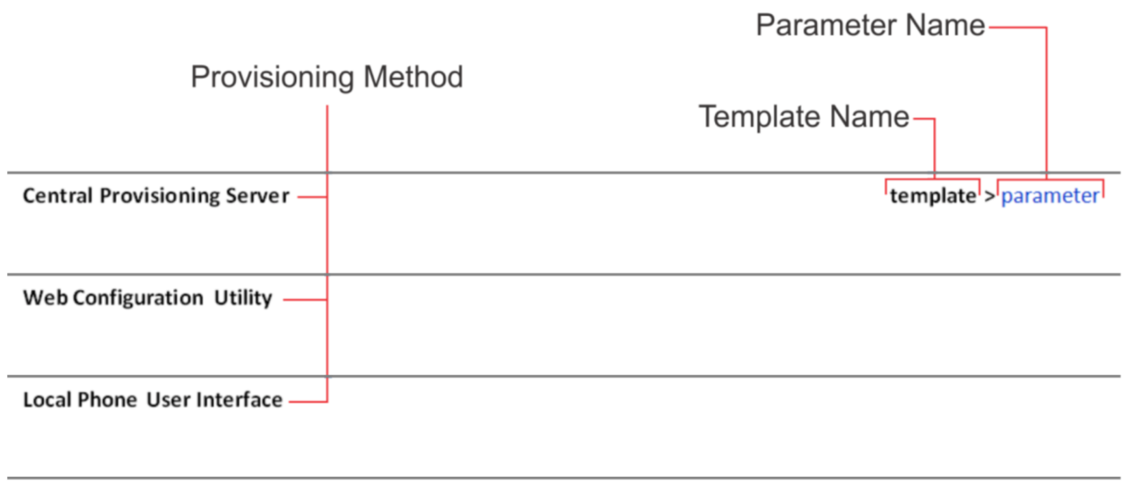
Read the Feature Parameter Tables

Each of the feature descriptions discussed in this administrator's guide includes a table of parameters that you configure to make the features work. This brief section explains the conventions used in the feature parameter tables. Polycom strongly recommends gaining familiarity with these conventions in order to read the tables and successfully perform configuration changes.

The feature parameter tables indicate one or more of three provisioning methods you can use to configure a feature: a centralized provisioning server, the Web Configuration Utility, or the local phone user interface. Note that the types of provisioning methods available for each feature will vary; not every feature uses all three methods.

The central provisioning server method requires you to configure parameters located in template configuration files that Polycom provides in XML format. The following illustration shows you how to use the parameter tables to locate the template name and the name of the parameter you configure to get the phone features working.

Feature Parameter Table Format



To quickly locate a specific parameter, locate and open the template name indicated. Then, use the parameter name to navigate the folders in the XML tree structure. The parameter name contains the XML folder path. The two following examples explain this convention in more detail.

Example One: Feature Parameter Tables

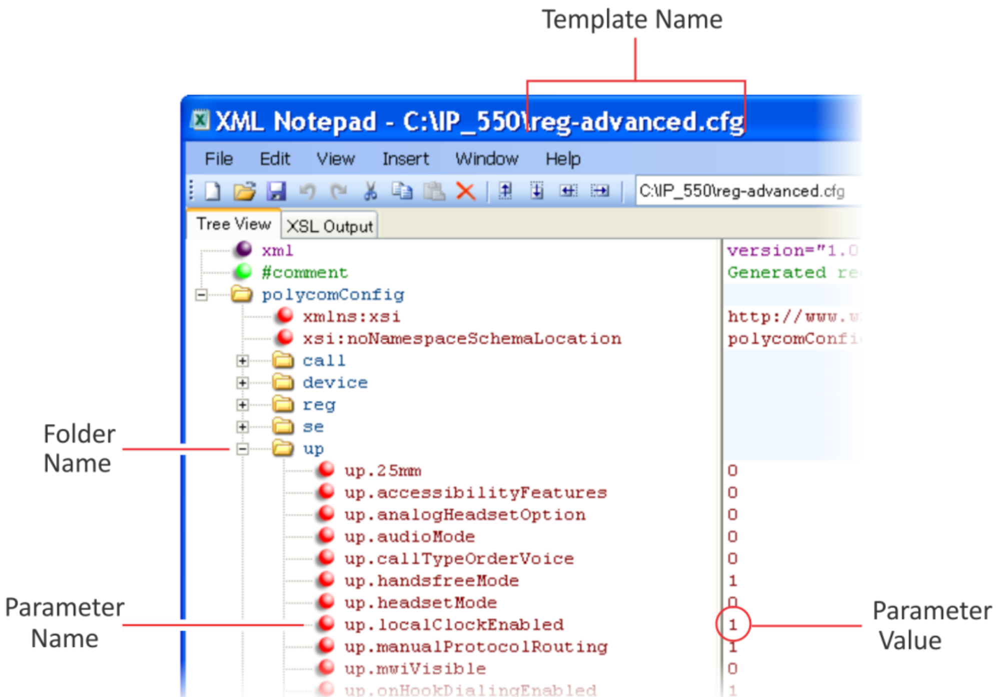
The example shown next is taken from the section Set the Time and Date Display in the Configuration Methods section.

Feature Parameter Table for Time and Date Display

Central Provisioning Server	template > parameter
Turn the time and date display on or off...	**reg-advanced.cfg** > up.localClockEnabled

This example indicates that the **reg-advanced.cfg** template file contains the `up.localClockEnabled` parameter, which turns the time and date display on or off. This parameter is enabled by default. If you want to turn the time and date display on or off, locate and open the **reg.advanced** template, expand the **up** folder, and locate the parameter name `up.localClockEnabled`. Set the parameter value to `1` to turn on or `0` to turn off the time and date display, as shown in the following illustration.

Example Time and Date Display



Note that some of the file paths in the templates are long and you may have to expand several folders in the XML tree structure to locate a specific parameter.

Note also that some feature parameters are located in more than one template file. In these cases, the parameter tables will list all related template files.

Tip: Each Parameter Is Linked

Each parameter listed in the tables in various sections is linked to its definition in the section Third-Party Software. The sections in that section define each parameter and list the permissible values, including the default value, of each parameter. If you want to find out more about a parameter you see listed in the tables, click the parameter name.

Example Two: Configuring Grouped Parameters

Some of the features have several related parameters that you must configure to get the feature working. In these cases, instead of listing every parameter, the table will specify a group of related parameters with an abbreviated XML path name ending with (.*), which indicates you can configure a group of related parameters.

Abbreviated XML paths, like full parameter names, are linked to their definitions in the reference sections in the section Third-Party Software. Specifically, since the reference sections lists parameters alphabetically, abbreviated XML path are linked to the first of a group of parameters listed alphabetically in the reference section. The next example shows you that in the **site.cfg** template, the `tcpIpApp.sntp` folder contains several related parameters that configure basic SNTP settings.

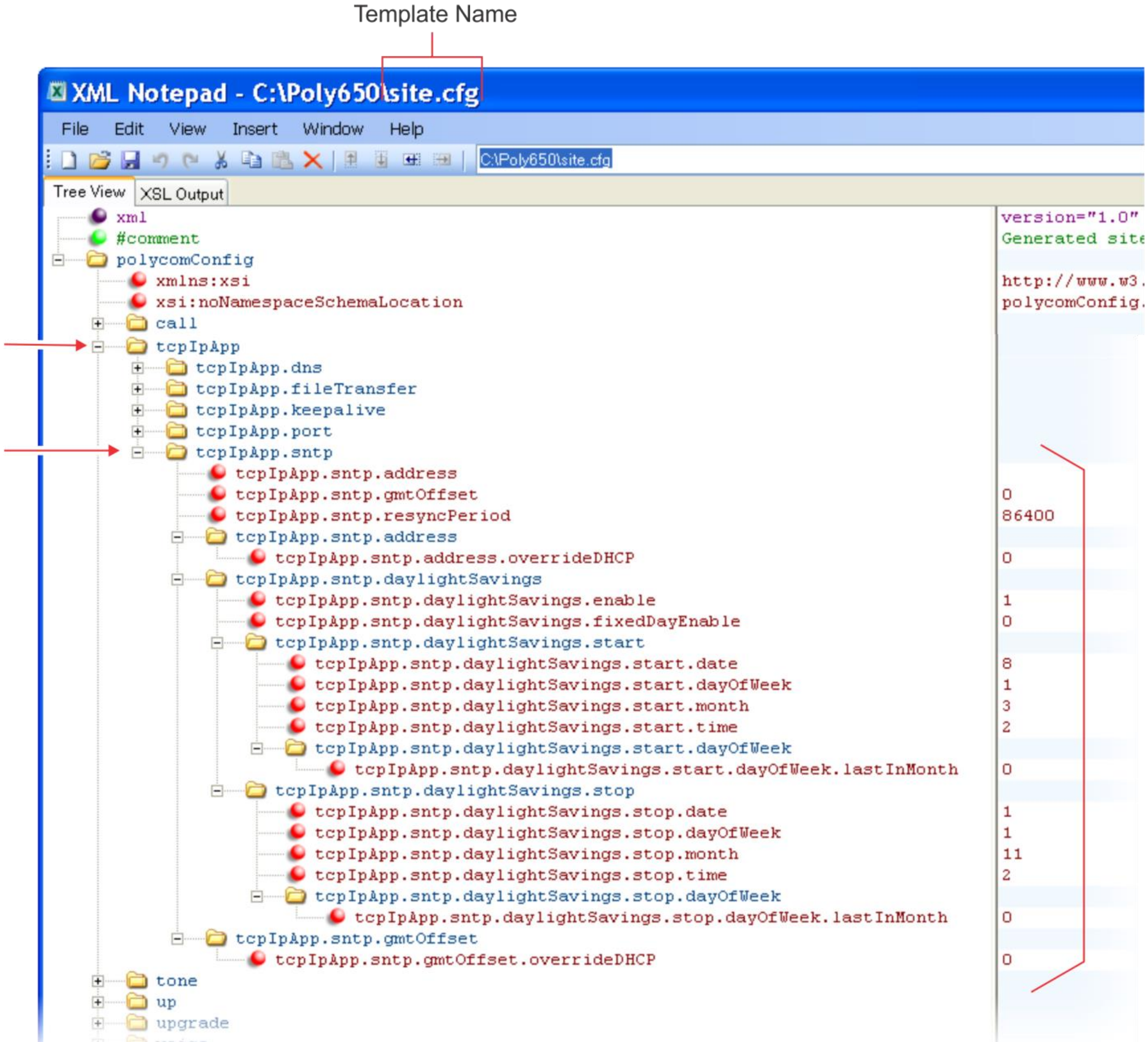
Feature Parameter Table for Time and Date SNTP Settings

Central Provisioning Server	template > parameter
Set the basic SNTP settings and daylight savings parameters	**site.cfg** > tcpIpApp.sntp.*

This example indicates that there is a group of SNTP parameters you can configure in the **site.cfg** template file. The abbreviated parameter name `tcpIpApp.sntp.*` indicates that you can configure parameters in the `tcpIpApp.sntp` folder as well as parameters in `tcpIpApp.sntp` subfolders.

To locate these parameters in the XML file, use the parameter name. The parameter name contains the XML folder path, as shown in the following illustration.

Locating Parameters in the Templates



In cases where the feature has several related parameters, you may find it helpful to refer to the parameter reference section in the section Polycom UC Software Menu System for a definition of each parameter. All parameter names, including abbreviated names, are linked to the parameter reference section - simply click on the parameter name.

This section has shown you how to read the configuration parameter tables so that you can locate the parameters in the XML template file.

Tip: Using an XML Editor

Polycom recommends using an XML editor such as XML Notepad 2007 to open and edit the configuration template files.

Get Help and Support

If you are looking for help or technical support for your phones, the following types of documents are available at the Polycom Support Center:

- Quick Start Guides, which describe how to assemble phones
- Quick User Guides, which describe the basic phone features
- User Guides, which describe both basic and advanced phone features
- Web Applications Developer's Guide, which provides guidance in the development of applications that run on your phone's Web browser or microbrowser
- Feature Description and Technical Notifications, such as Technical Bulletins and Quick Tips, that describe workarounds to existing issues and provide expanded descriptions and examples
- Release Notes, which describe the new and changed features and fixed problems in the latest version of the software

You can find Request for Comments (RFC) documents by entering the RFC number at http://www.ietf.org/rfc.html.

For other references, look for the Web Info icon  throughout this Administrator's Guide.

About the CX5500 Unified Conference Station for Microsoft Lync

You can use the Polycom CX5500 Unified Conference Station to make the following types of calls:

- Audio-only conference calls with Open SIP voice platforms or in a Lync Server environment.
- Audio and video calls made using Microsoft® Lync™. When your CX5500 system is connected to a computer running Lync client, the system provides a 360-degree view of the conference room and automatically identifies the active speaker.

Note that this administrator's guide focuses on configuring the telephony features available on the CX5500 system when used as an audio-only conference phone. For information on configuring settings available on the CX5500 system when connected to a computer and used as a audio and video conference phone, see the section Use the CX5100/5500 Control Panel.

The Polycom UC Software Big Picture

This section provides an overview of the Polycom UC software and the components that the CX5500 Unified Conference System uses in your network configuration.

The Polycom CX5500 Unified Conference Station supports most of the features of the Polycom UC Software 5.0.1 release. This administrator's guide describes the supported features; you can find additional helpful information about the UC Software 5.0.1 release at the Polycom United Communications Resource Center.

The UC software supports the deployment of Polycom phones as a Session Initiation Protocol (SIP)-based endpoint interoperating with a SIP call server or softswitch.

The Session Initiation Protocol (SIP) is the Internet Engineering Task Force (IETF) standard for multimedia communications over IP. It is an ASCII-based, application-layer control protocol, defined in RFC 3261, that you can use to establish, maintain, and terminate calls between two or more endpoints. Like other voice over IP (VoIP) protocols, SIP is designed to address the functions of signaling and session management within a packet telephony network. Signaling call information to be carried across network boundaries. Session management provides the ability to control the attributes of an end-to-end call.

For Polycom phones to successfully operate as a SIP endpoint in your network, you will require:

- A working IP network
- Routers configured for VoIP
- VoIP gateways configured for SIP
- The latest (or a compatible version) Polycom UC Software image
- An active, configured call server to receive and send SIP messages

  For information on IP PBX and softswitch vendors, see the Polycom ARENA VoIP Interoperability Partners list.

The rest of this section consists of the following sections:

- Understand Polycom UC Software Architecture

To begin setting up your Polycom phones on the network, go to Set Up Your Device Network.

To begin configuring the features available for your Polycom phones, start with the section Set Up Basic Phone Features.

Understand Polycom UC Software Architecture

This section discusses the components of the UC software.

What Is the Polycom UC Software?

The Polycom Unified Communications Software, or Polycom UC Software, manages the protocol stack, the digital signal processor (DSP), the user interface, and the network interaction. The UC Software implements the following functions and features on the phones:

- VoIP signaling for a wide range of voice and video telephony functions using SIP signaling for call setup and control
- SIP signaling
- Industry standard security techniques for ensuring that all provisioning, signaling, and media transactions are robustly authenticated and encrypted
- Advanced audio signal processing for speakerphone communications using a wide range of audio codecs
- Flexible provisioning methods to support single phone, small business, and large multi-site enterprise deployments

The software is a binary file image and contains a digital signature that prevents tampering or the loading of rogue software images.

There is a new image file in each release of software.

What Are the Configuration Files?

The Polycom UC Software that you download contains template configuration files, valid XML files that you can change using an XML editor. These template files contain a number of parameters that provision the phones with features and settings. The template configuration files are very flexible: you can rearrange the parameters within the template, move parameters to new files, or create your own configuration files from only those parameters you want. This flexibility is useful when you want to apply the same features and settings to a large number of phones. Use of the configuration files to provision the phones with features and settings is called the centralized provision method—the configuration files enable you to store a single set of configuration files on a central provisioning server and configure all of your phones to read the same set of files.

Polycom recommends that you configure phones using the centralized provisioning method. However, there are several methods you can use to configure the phones and you can use one or multiple methods in combination. Alternatively, you can configure individual phones using the phone's menu system, accessible through the local user interface, or using Web Configuration Utility.

What Are the Resource Files?

In addition to the software and configuration files, the phones may require resource files in order to use some of the advanced features.

Examples of resource files include:

- Language dictionaries
- Custom fonts
- Ringtones
- Contact directories

If you need to remove resource files from a phone at a later date—for example, if you are giving the phone to a new user—you will have to apply factory default settings to that phone.

Set Up Your Device Network

The Polycom CX5500 system operates on an Ethernet local area network (LAN). This section shows you several automated and manual ways to configure Polycom phones to operate in a LAN.

Connecting your Polycom phone to the LAN will initiate a startup sequence. Note that only step 1 is required and automatic (except for phones on a WLAN). Steps 2, 3, and 4 are optional as all these settings can be manually configured on the device. It is common to complete step 3 using a DHCP server within the LAN. The phone uses the following startup sequence:

1. The phone establishes network connectivity.

   Wired phones will establish a 10M/100M/1000M network link with an Ethernet switch device. Telephony will not function until this link is established. If the phone cannot establish a link to the LAN, an error message *Network link is Down* will display.

2. Apply appropriate security and Quality of Service (QoS) settings (optional).

   Assign the phone to a VLAN and/or 802.1X authentication.

3. Establish DHCP negotiation with the network and IP address, network addressing options, network gateway address, and time server.

4. Provision server discovery.

   To facilitate boot time, contacting the provisioning server is delayed until the phone is operational. You can also disable contacting the provisioning server, for example, to reduce the server load after a power failure.

After the provisioning server discovery is complete, the phone initiates the provisioning process described Set Up the Provisioning Server.

These steps are described in more detail in the following sections:

- Establish Link Connectivity
- Security and Quality of Service Settings
- IP Communication Settings
- Provisioning Server Discovery
- Phone Network Menus

Digest Authentication for Microsoft Internet Information Services

If you want to use digest authentication against the Microsoft Internet Information Services server:

Use Microsoft Internet Information Server 6.0 or later.

Digest authentication needs the user name and password to be saved in reversible encryption.

The user account on the server must have administrative privileges.

The wildcard must be set as MIME type; otherwise, the phone will not download *.cfg, *.ld and other required files. This is because the Microsoft Internet Information Server cannot recognize these extensions and will return a “File not found” error. To configure wildcard for MIME type, see IIS 6.0 does not serve unknown MIME types.

For more information, see Digest Authentication in IIS 6.0 on Microsoft TechNet.

Establish Link Connectivity

Wired devices will establish a connection to the LAN. If you want to change the phone’s configuration, do so prior to connecting the devices.

Wired Devices

Typical network equipment supports one of the three following Ethernet line rates: 10Mbps, 100Mbps, and 1000Mbps. The phones are configured to automatically negotiate the Ethernet rate so that no special configuration is required. You do have the option to change the line rates and/or duplex configuration. Polycom recommends that you keep the default settings. If you do change the settings, you should do so before deploying the phones.

Security and Quality of Service Settings

You have the option of using several layer-2 mechanisms that increase network security and minimize audio latency. This section describes each of the network security options.

VLANs and Wired Devices

A Virtual LAN (VLAN) can be used to separate and assign higher priority to a voice VLAN as a way of minimizing latency.

There are several methods in which the phone can be configured to work on a particular VLAN:

- **LLDP** Link Layer Discovery Protocol (LLDP) is a vendor-neutral Layer 2 protocol that allows a network device to advertise its identity and capabilities on the local network. To change these parameters, go to VLAN Menu.
- **CDP Compatible** Cisco Discovery Protocol (CDP) is a proprietary Data Link Layer network protocol. CDP Compatible follows the same set of rules. To change this parameter, go to VLAN Menu.

- **DHCP** Dynamic Host Configuration Protocol (DHCP) is an automatic configuration protocol used on IP networks. To change this parameter, go to DHCP Menu. To use DHCP for assigning VLANs, see Assign a VLAN ID Using DHCP. Note that the use of DHCP for assigning VLANs is not well standardized and is recommended only if the switch equipment does not support LLDP or CDP Compatible methods.
- **Static** The VLAN ID can be manually set from the phone UI or from a configuration file. To change this parameter, go to VLAN Menu. This will set the device setting parameter only.

If the phone receives a VLAN setting from several of the above methods, the priority is as follows (from highest to lowest):

- LLDP
- CDP
- Device settings
- DHCP VLAN discovery

802.1X Authentication

802.1X authentication is a technology that originated for authenticating Wi-Fi links. It has also been adopted for authenticating PCs within fixed LAN deployments.

When VoIP phones (with a secondary Ethernet port) are used to connect PCs on a network the 802.1X authentication process becomes more complex since the PC is not directly connected to the 802.1X switch.

Web Info: 802.1X References

For more information on 802.1X authentication, see Introduction to IEEE 802.1X and Cisco® Identity-Based Networking Services (IBNS) at Cisco 802.1X.
See also IEEE 802.1X Multi-Domain Authentication on Cisco Catalyst Layer 3 Fixed Configuration Switches Configuration Example.

There are several ways to configure 802.1X authentication of devices connected to the PC port of the phone:

- You can configure many switches to automatically *trust* or accept a VoIP phone based on its MAC address. This is sometimes referred to as MAC Address Bypass (MAB).
- Some switches support a feature whereby they will to automatically *trust* a device that requests a VLAN using the CDP protocol.
- Some deployments support Multiple Device Authentication (MDA). In this situation, both the phone and the PC will separately authenticate themselves.

  In this scenario since the phone is closest to the 802.1X switch, the phone needs to notify the switch when the PC is disconnected. This can be achieved using an 802.1X EAPOL-Logoff message.

All of these methods are supported by Polycom products.

To change these parameters, see the section 802.1X Menu.

IP Communication Settings

When the phone has established network connectivity it needs to acquire several IP network settings to proceed with provisioning. These settings are typically obtained automatically from a DHCP server.

Tip: For Novice Administrators

Read this section if you are new to this process or have never set up a provisioning server before.

You have the option to set the IP communication settings manually from the phone UI, or to pre-provision using a `device.set` capability.

When making the DHCP request the phone will include information in Option 60 that can assist the DHCP server in delivering the appropriate settings to the device. For more information, see *Technical Bulletin 54041: Using DHCP Vendor Identifying Options with Polycom Phones*.

Timesaver: Reducing Repetitive Data

Polycom recommends using DHCP where possible to eliminate repetitive manual data entry.

The table DHCP Network Parameters details the settings that are supported through the DHCP menu:

DHCP Network Parameters

Parameter	*DHCP Option*	*DHCP*	*DHCP INFORM*	*Configuration File (application only)*	*Device Settings*
IP address	-	•	-	-	•
Subnet mask	1	•	-	-	•
IP gateway	3	•	-	-	•
Boot server address	See DHCP Menu or Provisioning Server Discovery.	•	•	-	•
SIP server address	151 Note: You can change this value by changing the device setting. See <device/>.	•	-	-	•
SNTP server address	Look at option 42, then option 4.	•	-	•	•
SNTP GMT offset	2	•	-	•	•

Parameter	*DHCP Option*	*DHCP*	*DHCP INFORM*	*Configuration File (application only)*	*Device Settings*
DNS server IP address	6	•	-	-	•
DNS INFORM server IP address	6	•	-	-	•
DNS domain	15	•	-	-	•
VLAN ID	See DHCP Menu.	**Warning**: Link Layer Discovery Protocol (LLDP) overrides Cisco Discovery Protocol (CDP). CDP overrides Local FLASH which overrides DHCP VLAN Discovery.			

Web Info: RFC Information on DHCP Options

For more information on DHCP options, see RFC 2131 and RFC 2132.

Note: Overriding the DHCP Value

The configuration file value for **SNTP server address** and **SNTP GMT offset** can be configured to override the DHCP value: see tcpIpApp.sntp.address.overrideDHCP.
The CDP Compatibility value can be obtained from a connected Ethernet switch if the switch supports CDP.

If you do not have control of your DHCP server or do not have the ability to set the DHCP options, enable the phone to automatically discover the provisioning server address. One way is to connect to a secondary DHCP server that responds to DHCP INFORM queries with a requested provisioning server value. For more information, see RFC 3361 and RFC 3925.

Provisioning Server Discovery

After the phone has established network connectivity it proceeds to the Configuration stage. In this stage the following steps are carried out:

- Software update
- Application of configuration settings relevant to a customer network

Admin Tip: Setting Up a Provisioning Server

Read this section if you are new to this process or have never set up a provisioning server before.

In many deployments, a centralized provisioning server is used for the software update and configuration functions. The phone supports several methods to 'discover' this provisioning server:

- **Static** You can manually configure the server address from the phone's user interface or the Web Configuration Utility, or you can pre-provision the phone. The server address is manually configured from the phone's user interface, the Web Configuration Utility, or pre-provisioned using `device.set` in a configuration file.
- **DHCP** A DHCP option is used to provide the address or URL between the provisioning server and the phone.
- **DHCP INFORM** The phone makes an explicit request for a DHCP option (which can be answered by a server that is not the primary DHCP server). For more information, see RFC 3361 and RFC 3925.
- **Quick Setup** This feature offers a soft key to the user that takes them directly to a screen to enter the provisioning server address and information. This is simpler than navigating the menus to the relevant places to configure the provisioning parameters. For more information, see *Technical Bulletin 45460: Using Quick Setup with Polycom Phones*.

To change these parameters, go to Provisioning Server Menu.

Web Info: Provisioning Polycom Phones

For more information on best practices with respect to provisioning, see *White Paper 60806: UC Software Provisioning Best Practices*.

Supported Provisioning Protocols

Updating the software performs the provisioning functions of uploading log files, master configuration files, software updates, and device setting menu changes.

By default, phones are shipped with FTP enabled as the provisioning protocol. You can change the provisioning protocol by updating the *Server Type* option. Or, you can specify a transfer protocol in the *Server Address*, for example, *http://usr:pwd@server* (see Provisioning Server Menu). The Server Address can be an IP address, domain string name, or URL. It can be obtained through DHCP.

Configuration file names in the ***<MACaddress>*.cfg** file can include a transfer protocol, for example, https://usr:pwd@server/dir/file.cfg. If a user name and password are specified as part of the server address or file name, they will be used only if the server supports them. If a user name and password are required but not specified, the device settings are sent to the server.

Tip: Choosing a Valid URL

A URL should contain forward slashes instead of back slashes and should not contain spaces. Escape characters are not supported. If a user name and password are not specified, the Server User and Server Password from device settings will be used (see Provisioning Server Menu).

Note: Active and Passive FTP Methods

There are two types of FTP methods - active and passive. UC Software is not compatible with active FTP.

Note: HTTP/HTTPS Authentication

Both digest and basic authentication are supported when using HTTP/HTTPS for UC Software. Only digest authentication is supported when using HTTP by the Updater.

To guarantee software integrity, updating the software downloads only cryptographically signed UC Software images. For HTTPS, widely recognized certificate authorities are trusted by the phone and custom certificates can be added to the phone.

Web Info: To View Trusted Certificate Authorities

For more information, see the section Trusted Certificate Authority List and *Technical Bulletin 17877: Using Custom Certificates With Polycom Phones*.

As of SIP 3.2, Mutual Transport Layer Security (TLS) authentication is available. For more information, see Support Mutual TLS Authentication.

802.1X authentication is available. For more information, see Support 802.1X Authentication.

Digest Authentication for Microsoft Internet Information Services (IIS)

If you want to use digest authentication against the Microsoft Internet Information Services server:

Use Microsoft Internet Information Server 6.0 or later.

Digest authentication needs the user name and password to be saved in reversible encryption.

The user account on the server must have administrative privileges.

The wildcard must be set as MIME type; otherwise, the phone will not download *.cfg, *.ld and other required files. This is because the Microsoft Internet Information Server cannot recognize these extensions and will return a "File not found" error. To configure wildcard for MIME type, see IIS 6.0 does not serve unknown MIME types on Microsoft Support.

For more information, see Digest Authentication in IIS 6.0 on Microsoft TechNet.

Phone Network Menus

You have the option of modifying the phone network configuration.

Tip: For Novice Administrators

Read this section if you are new to this process or have never set up a provisioning server before.

You can update the network configuration parameters after your phone starts and is running CX5500 Software. The network configuration menu is accessible from the phone's main menu. Select **Settings > Advanced > Admin Settings > Network Configuration**. To access the Advanced menu, you will have to enter the administrator's password.

Tip: Changing the Default Administrator Password

Polycom recommends that you change the default administrative password. See Local User and Administrator Passwords.

You have the option of modifying the phone network configuration parameters in the following menus and sub-menus:

- Main Menu
- Provisioning Server Menu
- DHCP Menu
- Network Interfaces Menu (Ethernet Menu)
- VLAN Menu
- 802.1X Menu
- PAC File Information
- Login Credentials Menu
- TLS Security
- TLS Profile Menu
- Applications Menu
- Syslog Menu

Use the soft keys, the arrow keys, and the Select and Delete keys to make changes.

Certain parameters are read-only due to the value of other parameters. For example, if the **DHCP** client parameter is enabled, the **Phone IP Address** and **Subnet Mask** parameters are grayed out or not visible since the DHCP server automatically supplies these parameters and the statically assigned IP address and subnet mask will never be used in this configuration.

Tip: Resetting Network Configurations

The basic network configuration referred to in the subsequent sections can be reset to factory default settings using the phone's main menu: Select **Settings > Advanced > Admin Settings > Reset to Defaults > Reset Device Settings**.

Main Menu

You can modify the configuration parameters shown in the table Main Menu from the setup menu while the phone boots, or from the Administrative Settings menu from a phone running CX5500 Software.

Main Menu

Name	*Possible Values*
Provisioning Menu	
See Provisioning Server Menu.	
Network Interfaces Menu or Ethernet Menu	
See Network Interfaces Menu (Ethernet Menu).	
TLS Security Menu	
See TLS Security Menu.	
SNTP Address	**Dotted-decimal IP address OR Domain name string**
The Simple Network Time Protocol (SNTP) server the phone obtains the current time from.	
GMT Offset	**-13 through +12**
The offset of the local time zone from Greenwich Mean Time (GMT) in half hour increments.	
DNS Server	**Dotted-decimal IP address**
The primary server the phone directs Domain Name System (DNS) queries to.	
DNS AltServer	**Dotted-decimal IP address**
The secondary server to which the phone directs DNS queries.	
DNS Domain	**Domain name string**
The phone's DNS domain.	
Hostname	**hostname**
The DHCP client hostname.	
Syslog Menu	
See Syslog Menu.	

Name	*Possible Values*
Quick Setup	**Enabled, Disabled**
If enabled, a QSetup soft key displays on the idle screen when you are in Lines View. When you tap this soft key, a menu displays enabling you to configure the parameters required to access the provisioning server. **Note**: The Quick Setup option is not available in the Updater.	
Reset to Defaults	
There are five ways to reset or clear phone features and settings, including settings from web or local override files.	
Base Profile	**Generic, Lync**
Use this to enable Lync-compatible phones to register with Lync Server. When set to Lync, the phone automatically provisions with the minimum parameters required to register with Lync Server. You cannot modify or customize the Base Profile. By default, the Base Profile is set to Generic.	

Settings: Preventing Invalid Parameter Values

If you insert incorrect parameter values into the configuration file, the phone ignores the invalid values and uses the previous configuration. Before you complete your configuration, make sure you set values for these parameters.

Provisioning Server Menu

The configuration parameters shown in Provisioning Server Menu can be modified on the Provisioning Server Menu.

Provisioning Server Menu

Name	*Possible Values*
DHCP Menu	
See DHCP Menu. **Note**: This menu is disabled when the DHCP client is disabled.	
Server Type	**0=FTP, 1=TFTP, 2=HTTP, 3=HTTPS, 4=FTPS**
The protocol that the phone uses to obtain configuration and phone application files from the provisioning server. See Supported Provisioning Protocols. **Note**: Active FTP is not supported for BootROM version 3.0 or later. Passive FTP is supported. Only implicit FTPS is supported.	

Name	*Possible Values*
Server Address	**Dotted-decimal IP address OR URL**
Domain name string or a URL. All addresses can be followed by an optional directory. The address can also be followed by the file name of a **.cfg** master configuration file, which the phone will use instead of the default **<MACaddress>.cfg** file. The provisioning server to use if the DHCP client is disabled, if the DHCP server does not send a boot server option, or if the **Boot Server** parameter is set to **Static**. The phone can contact multiple IP addresses per DNS name. These redundant provisioning servers must all use the same protocol. If a URL is used, it can include a user name and password. See Supported Provisioning Protocols. For information on how to specify a directory and use the master configuration file, see Understand the Master Configuration File. **Note**: ":", "@", or "/" can be used in the user name or password if they are correctly escaped using the method specified in RFC 1738.	
Server User	**String**
The user name requested when the phone logs into the server (if required) for the selected **Server Type**. **Note**: If the Server Address is a URL with a user name, this will be ignored.	
Server Password	**String**
The password requested when the phone logs in to the server if required for the selected **Server Type**. **Note**: If the Server Address is a URL with user name and password, this will be ignored.	
File Transmit Tries	**1 to 10 Default 3**
The maximum number of attempts to transfer a file. (An attempt is defined as trying to download the file from all IP addresses that map to a particular domain name.)	
Retry Wait	**0 to 300 seconds Default 1**
The minimum amount of time that must elapse before retrying a file transfer. The time is measured from the start of a transfer attempt, which is defined as the set of upload/download transactions made with the IP addresses that map to a given provisioning server’s DNS. If the set of transactions in an attempt is equal to or greater than the Retry Wait value, then there will be no further delay before the next attempt is started. For more information, see Deploy and Update the CX5500 System with a Provisioning Server.	
Tag SN to UA	**Disabled, Enabled**
If enabled, the phone’s serial number (MAC address) is included in the User-Agent header of HTTP/HTTPS transfers and communications to the browser. The default value is Disabled.	
Upgrade Server	**String**
The address/URL that will be accessed for software updates requested from the phones Web configuration utility.	
ZTP	**Disabled, Enabled**
See Zero-Touch Provisioning Solution on Polycom Voice Support.	

Tip: Changing the Default Passwords

The Server User and Server Password parameters should be changed from the default values.

DHCP Menu

The DHCP menu is accessible only when the DHCP client is enabled. You can update DHCP configuration parameters shown in the table DHCP Menu.

DHCP Menu

<table>
<tr><th>Name</th><th>Possible Values</th></tr>
<tr><td>Boot Server</td><td>0=Option 66, 1=Custom, 2=Static, 3=Custom+Option 66</td></tr>
<tr><td colspan="2">
Option 66 The phone will look for option number 66 (string type) in the response received from the DHCP server. The DHCP server should send address information in option 66 that matches one of the formats described for Server Address in Provisioning Server Menu.
Custom The phone will look for the option number specified by the Boot Server Option parameter (below), and the type specified by the Boot Server Option Type parameter (below) in the response received from the DHCP server.
Static The phone will use the boot server configured through the Server Menu. For more information, see Provisioning Server Menu.
Custom + Option 66 The phone will use the custom option first or use Option 66 if the custom option is not present.
Note: If the DHCP server sends nothing, the following scenarios are possible:

If a boot server value is stored in flash memory and the value is not <code>0.0.0.0</code>, then the value stored in flash is used.
Otherwise the phone sends out a DHCP INFORM query.

If a single DHCP INFORM server responds, this is functionally equivalent to the scenario where the primary DHCP server responds with a valid boot server value.
If no DHCP INFORM server responds, the INFORM query process will retry and eventually time out.

If the server address is not discovered using DHCP INFORM then the phone will contact the ZTP server if the ZTP feature is enabled.

</td></tr>
<tr><td>Boot Server Option</td><td>128 through 254 (Cannot be the same as VLAN ID Option)</td></tr>
<tr><td colspan="2">When the Boot Server parameter is set to Custom, this parameter specifies the DHCP option number in which the phone will look for its boot server.</td></tr>
<tr><td>Boot Server Option Type</td><td>0=IP Address, 1=String</td></tr>
<tr><td colspan="2">When the Boot Server parameter is set to Custom, this parameter specifies the type of DHCP option in which the phone will look for its provisioning server. The IP Address provided must specify the format of the provisioning server. The String provided must match one of the formats described for Server Address in Provisioning Server Menu.</td></tr>
<tr><td>Option 60 Format</td><td>0=RFC 3925 Binary, 1=ASCII String</td></tr>
<tr><td colspan="2">
RFC 3925 Binary: Vendor-identifying information in the format defined in RFC 3925.
ASCII String: Vendor-identifying information in ASCII.
For more information, see Technical Bulletin 54041: Using DHCP Vendor Identifying Options With Polycom Phones.
Note: DHCP option 125 containing the RFC 3295 formatted data will be sent whenever option 60 is sent. DHCP option 43 data is ignored.
</td></tr>
</table>

Note: Multiple DHCP INFORM Servers

If multiple DHCP INFORM servers respond, the phone should gather the responses from these DHCP INFORM servers. If configured for Custom+Option66, the phone will select the first response that contains a valid *custom* option value. If none of the responses contain a *custom* option value, the phone will select the first response that contains a valid *option66* value.

Network Interfaces Menu (Ethernet Menu)

The Network Interfaces Menu displays only if there are multiple network interfaces to the phone. For the CX5500 system, the Ethernet menu displays instead of the Network Interfaces menu.

You can select the following items in the Network Interfaces Menu:

- Ethernet Menu

You can select items shown in the table Ethernet Menu.

Ethernet Menu

Name	*Possible Values*
DHCP	**Enabled, Disabled**
If enabled, DHCP will be used to obtain the parameters discussed in IP Communication Settings.	
IP Address	**Dotted-decimal IP address**
The phone's IP address. **Note**: This parameter is disabled when DHCP is enabled.	
Subnet Mask	**Dotted-decimal subnet mask**
The phone's subnet mask. **Note**: This parameter is disabled when DHCP is enabled.	
IP Gateway	**Dotted-decimal IP address**
The phone's default router.	
VLAN	
See VLAN Menu.	
802.1X Authentication	**Enabled, Disabled**
If enabled, the phone will use the 802.1 Authentication parameters to satisfy the negotiation requirements for each EAP type.	
802.1X Menu	
See 802.1X Menu.	
LAN Port Mode	**0 = Auto, 1 = 10HD, 2 = 10FD, 3 = 100HD, 4 = 100FD, 5 = 1000FD**
The network speed over Ethernet. The default value is Auto. HD means half duplex and FD means full duplex. Note: Polycom recommends that you do not change this setting.	

<table>
<tr><th>Name</th><th>Possible Values</th></tr>
<tr><td>PC Port Mode</td><td>0 = Auto, 1 = 10HD, 2 = 10FD, 3 = 100HD, 4 = 100FD, 5 = 1000FD, -1 = Disabled</td></tr>
<tr><td colspan="2">The network speed over Ethernet. The default value is Auto. HD means half duplex and FD means full duplex.
Note: Polycom recommends that you do not change this setting unless you want to disable the PC port.</td></tr>
<tr><td>1000BT LAN Clock</td><td>0=Auto 1=Slave 2=Master</td></tr>
<tr><td colspan="2">The mode of the LAN clock.
The default value is Slave (this device receives its clock timing from a master device).
Note: Polycom recommends that you do not change this setting unless you have Ethernet connectivity issues. This setting was chosen to give the best results from an EMI perspective.</td></tr>
<tr><td>1000BT PC Clock</td><td>0=Auto 1=Slave 2=Master</td></tr>
<tr><td colspan="2">The mode of the PC clock. The default value is Auto.
Note: Polycom recommends that you do not change this setting unless you have Ethernet connectivity issues. This setting was chosen to give the best results from an EMI perspective.</td></tr>
</table>

VLAN Menu

You can modify the parameters shown in the table VLAN Menu.

VLAN Menu

<table>
<tr><th>Name</th><th>Possible Values</th></tr>
<tr><td>VLAN ID</td><td>Null, 0 through 4094</td></tr>
<tr><td colspan="2">The phone’s 802.1Q VLAN identifier. The default value is Null.
Note: Null = no VLAN tagging</td></tr>
<tr><td>LLDP</td><td>Enabled, Disabled</td></tr>
<tr><td colspan="2">If enabled, the phone will use the LLDP protocol to communicate with the network switch for certain network parameters. Most often this will be used to set the VLAN that the phone should use for voice traffic. It also reports power management to the switch. The default value is Enabled.
For more information on how to set VLAN and LLDP, see LLDP and Supported TLVs.</td></tr>
<tr><td>CDP Compatibility</td><td>Enabled, Disabled</td></tr>
<tr><td colspan="2">If enabled, the phone will use CDP-compatible signaling to communicate with the network switch for certain network parameters. Most often this will be used to set the VLAN that the phone should use for Voice Traffic, and for the phone to communicate its PoE power requirements to the switch. The default value is Enabled.</td></tr>
<tr><td>VLAN Discovery</td><td>0=Disabled, 1=Fixed (default), 2=Custom</td></tr>
<tr><td colspan="2">For a detailed description, see Assign a VLAN ID Using DHCP.
Disabled: No VLAN discovery through DHCP.
Fixed: Use predefined DHCP vendor-specific option values of 128, 144, 157 and 191. If one of these is used, VLAN ID Option will be ignored
Custom: Use the number specified for VLAN ID Option as the DHCP private option value.</td></tr>
</table>

Name	*Possible Values*
VLAN ID Option	**128 through 254 (Cannot be the same as Boot Server Option) (default is 129)**
The DHCP private option (when VLAN Discovery is set to Custom). For more information, see Assign a VLAN ID Using DHCP.	

802.1X Menu

The 802.1X Menu displays when 802.1X authentication is enabled. You can modify configuration parameters shown in the 802.1X Menu.

802.1X Menu

Name	*Possible Values*
EAP Method	**0 = None, 1=EAP-TLS, 2=EAP-PEAPv0/MSCHAPv2, 3=EAP-PEAPv0/GTC, 4=EAP-TTLS/EAP-MSCHAPv2, 5=EAP-TTLS/EAP-GTC, 6=EAP-FAST, 7=EAP-MD5**
The selected EAP type to be used for authentication. For more information, see Support 802.1X Authentication.	
Identity	**UTF-8 encoded string**
The identity (or user name) required for 802.1X authentication.	
Password	**UTF-8 encoded string**
The password required for 802.1X authentication. The minimum length is 6 characters.	
Anonymous ID	**UTF-8 encoded string**
The anonymous user name for constructing a secure tunnel for tunneled authentication and FAST authentication.	
PAC File Info	
See PAC File Information.	
EAP-FAST Inband Provisioning	**Enabled, Disabled**
A flag to determine whether EAP-FAST Inband Provisioning is enabled. This parameter is used only if EAP Method is EAP-FAST.	

PAC File Information

You can modify Protected Access Credential (PAC) File Information shown in the table PAC File Information Menu.

PAC File Information Menu

Name	*Possible Values*	*Description*
PAC File Password	**UTF-8 encoded string**	**The password required to decrypt the PAC file.**
PAC File Name	UTF-8 encoded string	The path or URL of the PAC file for download.
Remove PAC File	**UTF-8 encoded string**	**A flag to determine whether or not to delete the PAC file from the phone.**

Login Credentials Menu

You can modify the parameters shown in the Login Credentials Menu.

Login Credentials Menu

Name	*Possible Values*
Domain	**UTF-8 encoded string**
The domain name used by a server.	
User	**UTF-8 encoded string**
The user name used to authenticate to a server.	
Password	**UTF-8 encoded string**
The password used to authenticate to a server.	

TLS Security Menu

This section refers to the TLS Security menu available in the software. You can modify the parameters shown in the table TLS Security Menu.

TLS Security Menu

Name	*Possible Values*
OCSP	**Enabled, Disabled**
The Online Certificate Status Protocol checks the revocation status of X.509 digital certificates downloaded during negotiation of a TLS connection.	

Name	*Possible Values*
FIPS	**Enabled, Disabled**
The Federal Information Processing Standard enables the validation and usage of FIPS-140 certified encryption algorithms.	
Install Custom CA Cert	**URL**
A CA certificate that is installed on the phone to be used for TLS authentication.	
Install Custom Device Cert	**URL**
A device certificate installed on the phone to be used for Mutual TLS authentication.	
Clear Certificate	**Yes, No**
A flag to determine whether or not the device certificate can be removed from the phone.	
TLS Profile x	
There are currently two TLS Platform profiles. See TLS Profile Menu.	
Applications	
See Applications Menu.	

TLS Profile Menu

You can modify the parameters shown in table TLS Profile Menu.

TLS Profile Menu

Name	*Possible Values*
SSL Cipher Suite	**String**
The global cipher suite.	
Custom SSL Cipher Suite	**String**
A custom cipher suite.	
CA Cert List	**String**
The CA certificate sources that are valid for this profile.	
Device Cert List	**String**
The device certificate sources that are valid for this profile.	

Applications Menu

You can modify the parameters shown in the Applications Menu.

Applications Menu

Name	*Possible Values*
802.1X	**1 or 2**
The TLS Profile to use for 802.1X authentication.	
Provisioning	**1 or 2**
The TLS Profile to use for provisioning authentication.	
Provisioning	**Enable or Disable**
The TLS Profile to enable or disable common name validation.	
Syslog	**1 or 2**
The TLS Profile to use for syslog authentication.	

Syslog Menu

Syslog is a standard for forwarding log messages in an IP network. The term *syslog* is often used for both the actual syslog protocol as well as the application or library sending syslog messages.

The syslog protocol is a simple protocol: the syslog sender sends a small textual message (less than 1024 bytes) to the syslog receiver. The receiver is commonly called *syslogd*, *syslog daemon*, or *syslog server*. Syslog messages can be sent through UDP, TCP, or TLS. The data is sent in cleartext.

Because syslog is supported by a wide variety of devices and receivers, syslog can be used to integrate log data from many different types of systems into a central repository.

Web Info: Information on Syslog
For more information on the syslog protocol, see RFC 3164.

You can modify the parameters shown in the table Syslog Menu.

Syslog Menu

Name	*Possible Values*
Server Address	**Dotted-decimal IP address OR Domain name string**
The syslog server IP address. The default value is Null.	

Name	*Possible Values*
Server Type	**None=0, UDP=1, TCP=2, TLS=3**
The protocol that the phone will use to write to the syslog server. If set to None (or 0), transmission is turned off, but the server address is preserved.	
Facility	**0 to 23**
A description of what generated the log message. For more information, see section 4.1.1 of RFC 3164. The default value is 16, which maps to local 0.	
Render Level	**0 to 6**
Specifies the lowest class of event that will be rendered to syslog. It is based on `log.render.level` and can be a lower value. See <log/>. Note: Use left and right arrow keys to change values.	
Prepend MAC Address	**Enabled, Disabled**
If enabled, the phone’s MAC address is prepended to the log message sent to the syslog server.	

Set Up the Provisioning Server

This section provides instructions for setting up your system with a provisioning server. If you are new to this process, it is important to read every section in this section.

This section focuses on one particular way that the Polycom® CX5500 Software and the required external systems might initially be installed and configured in your network.

Set Up the Provisioning Server consists of the following sections:

- Why Use a Provisioning Server?
- Provisioning Server Security Notes
- Set up an FTP Server as Your Provisioning Server
- Download Polycom CX5500 Software Files to the Provisioning Server
- Deploy and Update the CX5500 System with a Provisioning Server
- Upgrade Polycom UC Software

Why Use a Provisioning Server?

Read this section if you have never set up a provisioning server before.

Polycom strongly recommends that you use a provisioning server to install and maintain your Polycom phones. You can set up a provisioning server on the local LAN or anywhere on the Internet. A provisioning server maximizes the flexibility you have when installing, configuring, upgrading, and maintaining the phones, and enables you to store configuration, log, directory, and override files on the server. If you allow the phone write access to your provisioning server, the phone can use the server to upload all of the file types and store administrator and user settings. The phone is designed such that if it cannot locate a provisioning server when it boots up, it will operate with internally saved parameters. This is useful when the provisioning server is not available.

Web Info: Registering Standalone Polycom Phones

If you want to register a single CX5500 system, see the *Polycom Web Configuration Utility User Guide*.

You can configure multiple (redundant) provisioning servers—one logical server with multiple addresses—by mapping the provisioning server DNS name to multiple IP addresses. The default number of provisioning servers is one and the maximum number is eight. For more information on the protocol used, see Supported Provisioning Protocols.

If you set up multiple provisioning servers, you must be able to reach all of the provisioning servers with the same protocol and the contents on each provisioning server must be identical. The parameters described in Provisioning Server Menu can be used to configure the number of times each server will be tried for a file transfer and also how long to wait between each attempt. You can configure the maximum number of servers to be tried. For more information, contact your Certified Polycom Reseller.

Provisioning Server Security Notes

Read this section if you have never set up a provisioning server before.

For organizational purposes, Polycom recommends configuring a separate log file directory, an override directory, a contact directory, and a license directory, though this is not required. Each directory can have different access permissions. For example, you can allow LOG, CONTACTS, and OVERRIDES to have full read and write access, and LICENSE to have read-only access.

You should ensure that the file permissions you create provide the minimum required access and that the account has no other rights on the server.

Tip: Allowing File Uploads to Your Provisioning Server

Polycom recommends that you allow file uploads to the provisioning server where the security environment permits. File uploads allow event log files to be uploaded. Log files provide backup copies of changes users make to the directory, and to the phone’s configuration through the Web server and/or local user interface. These log files help Polycom provide customer support when diagnosing issues that may occur with the phone operation.

The phone’s server account needs to be able to add files that it can write to in the log file directory and the provisioning directory. It must also be able to list files in all directories mentioned in the ***<MACaddress>*.cfg** file. All other files that the phone needs to read, such as the application executable and the standard configuration files, should be made read-only using file server file permissions.

Tip: Use RFC-Compliant Servers

Polycom recommends that you use RFC-compliant servers.

Each phone may open multiple connections to the server.

The phone will attempt to upload log files, a configuration override file, and a directory file to the server if changed. This requires that the phone’s account has delete, write, and read permissions. The phone will still function without these permissions, but will not be able to upload files.

If you know the phone is going to download a file from the server, you should mark the file as read-only.

Set up an FTP Server as Your Provisioning Server

Read this section if you have never set up a provisioning server before.

A simple provisioning configuration uses File Transfer Protocol or FTP. FTP servers are free, require installation, and use logins and passwords. A free and popular server, FileZilla Server, is available for Windows. FileZilla Server (version 0.9.xx) has been tested with the CX5500 Software.

Tip: Choosing a Provisioning Protocol

By default, Polycom sets FTP as the provisioning protocol on all Polycom phones. This guide focuses on the FTP provisioning protocol. Other supported protocols include TFTP, HTTP, and HTTPS.

To set up an FTP server using FileZilla Server:

1 Download and install the latest version of FileZilla Server.

2 After installation, a *Connect to Server* pop-up displays on your computer. Click **OK** to open the administrative user interface.

3 To configure a user, select **Edit > Users** in the status bar.

4 Click **Add.**

5 Enter the user name for the phone and select **OK**.

   For example, *bill123*.

6 Select the **Password** checkbox and enter a password.

   For example, *1234*. The phone will use this password to log in.

7 Select **Page > Shared folders** to specify the server-side directory where the provisioning files will be located (and the log files uploaded).

8 Select **Add** and pick the directory.

9 To allow the phone to upload logs onto the provisioning server, select **Shared Folders > Files**, then select **Write** and **Delete** checkboxes, and then click **OK**.

10 Determine the IP address of the FTP server by entering **cmd** in the **Run** dialog on your **Start** menu, and enter **ipconfig** in the command prompt.

   The IP Address of the FTP server is shown.

Download Polycom CX5500 Software Files to the Provisioning Server

This section explains how to download the Polycom Unified Communications (UC) Software to the provisioning server.

Go to the Polycom UC Software Support Center to download current and past releases, access supporting documentation, navigate to the downloads and documents available for a specific product. Polycom provides the UC Software download in .tar file format.

You can find the latest software for the CX5500 system under Downloads on the CX5500 Support page.

To download the Polycom UC Software for the CX5500 system:

1 On the CX5500 Support page, click the latest version of software available.

2 Click **Save** to download the software package.

3 Open and extract (uncompress) the .tar file.

4 Copy all files from the distribution .tar file to the home directory on the provisioning server, maintaining the same folder hierarchy. To simplify provisioning, Polycom recommends editing copies of each file as a best practice to ensure that you have unedited template files containing the default values.

- The split image file contains individual files for the CX5500 system as well as all of the template configuration files included in the combined image file.

For a list and brief description of all available template files included with Polycom CX5500 Software , see Use the Template Configuration Files.

Note: See the Release Notes for a Description of all Parameters for a UC Software Release

For a description of each file in a UC Software distribution, see the *UC Software Release Notes* for a particular UC Software release on the Polycom UC Software Support Center.

Deploy and Update the CX5500 System with a Provisioning Server

This section explains how to deploy and update the CX5500 system from a provisioning server. If you are provisioning the system using a provisioning server for the first time, follow the provisioning process shown in the section Deploy CX5500 Systems with a Provisioning Server. If you are using the systemin one of the following special scenarios, refer to the relevant section:

The CX5500 system can boot up without any configuration files; however, you must configure certain parameters in the configuration files - for example, a registration address, label, and SIP server address – to use the system.

You can create as many configuration files as you want and your configuration files can contain any combination of parameters. You can put all parameters into one file or, for example, you can put SIP server parameters in one file and phone features parameters in another file. For detailed information on how to use the configuration files, see the section Use the Template Configuration Files.

For large-scale deployments, the centralized provisioning method using configuration files is strongly recommended. For smaller scale deployments, the Web Configuration Utility or local interface can be used, but administrators need to be aware that settings made using these methods can override settings made using configuration files.

For instructions on how to encrypt your configuration files, see Encrypt Configuration Files.

Deploy CX5500 Systems with a Provisioning Server

You can deploy a group of CX5500 systems using the provisioning server.

To deploy the system with a provisioning server:

1 Obtain a list of MAC addresses for the phones you want to deploy.

The MAC address is a 12-digit hexadecimal number on a label on the back of the phone and on the outside of the shipping box.

2 Create a per-phone **phone<*MACaddress*>.cfg** file.

Tip: Choosing the File Name for a Per-Phone Configuration File

Do not use the following file names as your per-phone file name: <*MACaddress*>-phone.cfg, <*MACaddress*>-web.cfg, <*MACaddress*>-app.log, or <*MACaddress*>-license.cfg. These file names are used by the phone itself to store user preferences (overrides) and logging information.

3 Add the phone registration parameters to the file, for example `reg.1.address`, `reg.1.label`, and `reg.1.type`.

4 Create a per-site **site<*location*>.cfg** file.

For example, add the SIP server or feature parameters such as `voIpProt.server.1.address` and `feature.corporateDirectory.enabled`.

Settings: Configuring Your Phone for Local Conditions

Most of the default settings are typically adequate; however, if SNTP settings are not available through DHCP, edit the SNTP GMT offset, and possibly the SNTP server address for the correct local conditions. Changing the default daylight savings parameters will likely be necessary outside of North America. Disable the local Web (HTTP) server or change its signaling port if the local security policy dictates (see <httpd/>). Change the default location settings for user interface language and time and date format (see <lcl/>)

5 Create a master configuration file by performing the following steps:

a Enter the name of each per-phone and per-site configuration files created in steps 2 and 3 in the `CONFIG_FILES` attribute of the master configuration file (**000000000000.cfg**). For help using the master configuration file, see Understand the Master Configuration File.

For example, add a reference to **phone<*MACaddress*>.cfg**.

b (Optional) Edit the `LOG_FILE_DIRECTORY` attribute of master configuration file so that it points to the log file directory.

c (Optional) Edit the `CONTACT_DIRECTORY` attribute of master configuration file so that it points to the organization’s contact directory.

d (Optional) Edit the `USER_PROFILES_DIRECTORY` attribute of master configuration file if you intend to enable the User Login feature.

For more information, see Set User Profiles.

e (Optional) Edit the `CALL_LISTS_DIRECTORY` attribute of master configuration file so that it points to the user call lists.

6 Perform the following steps to configure the phone to point to the IP address of the provisioning server and set up the user:

a On the phone's Home scree, select **Settings > Advanced > Admin Settings > Network Configuration > Provisioning Server**.

   When prompted for the administrative password, enter **456**. The Provisioning Server entry is highlighted.

b Tap the **Select** soft key.

c Scroll down to **Server Type** and ensure that it is set to **FTP**.

d Scroll down to **Server Address** and enter the IP address of your provisioning server.

e Tap the **Edit** soft key to edit the value and the **OK** soft key to save your changes.

f Scroll down to **Server User** and **Server Password** and enter the user name and password of the account you created on your provisioning server, for example, *bill1234* and *1234,* respectively.

g Tap the **Back** soft key twice.

h Tap **Save & Reboot**.

   The phone reboots.

   At this point, the phone sends a DHCP Discover packet to the DHCP server. This is found in the Bootstrap Protocol/option "Vendor Class Identifier" section of the packet and includes the phone's part number and the BootROM version.

   For more information, see Parse Vendor ID Information.

7 Ensure that the configuration process completed correctly.

8 On the phone, select **Status > Platform > Application > Main** to see the UC Software version and **Status > Platform > Configuration** to see the configuration files downloaded to the phone.

9 Monitor the provisioning server event log and the uploaded event log files (if permitted). All configuration files used by the provisioning server are logged.

   You can now instruct your users to start making calls.

Upgrade Polycom UC Software

You can upgrade the software that is running on the CX5500 system in your organization. The UC Software executable and configuration files can all be updated using centralized provisioning.

You can also update the software for a single CX5500 system by placing a software repository on a USB thumb-drive, external hard-disk drive, or other type of USB storage media to update the system. When a flash drive is attached, the system scans the drive for a software repository – if a valid, different software update file is found, a notification displays enabling you to choose to apply or cancel the update. If you do not cancel within 30 seconds, the update begins automatically.

To update your software using a USB drive:

1 Format a USB flash drive as FAT32.

   If you are using a drive that is already formatted, ensure that previous software updates are deleted from the USB drive.

2 Download the software package to the USB drive. Update files have a .tar extension.

3. Connect the USB flash drive to the USB port on the tabletop unit or on the power data box.
4. On the CX5500 system, choose to apply the software update request displayed on the LCD screen.

   The system detects the new software on the USB drive and starts the update within 30 seconds. The indicator lights begin to flash, indicating that the update has started. The system reboots up to four times during the update, and the indicator lights flash in several different patterns.

   The update is complete when the indicator lights stop flashing.

Additionally, you can use the Web Configuration Utility to set up automatic software updates for a single CX5500 system. Note that configuration changes made to individual systems using the Web Configuration Utility overrides configuration settings made using central provisioning. For instructions on how to update UC Software using the Web Configuration Utility, see *Feature Profile 67993: Use the Software Upgrade Tool in the Web Configuration Utility*.

Configuration Methods

This section explains configuration methods you can use to configure settings and features on the phones:

- Local phone user interface (*for a single phone*)
- Web Configuration Utility (*for a single phone*)
- Centralized provisioning method (*for multiple phones*)

The methods explained in this section configure many of the phone features and settings detailed in this administrator's guide. Note that not all of the features and settings are available using each configuration method. You can use a single method or you can use a combination of methods depending on your preferences and your corporate security.

Web Info: Registering a Single Polycom Phone

If you want to register a single Polycom phone, see *Quick Tip 44011: Registering Standalone Polycom SoundPoint IP, SoundStation IP, and VVX 1500 Phones*.

Polycom recommends using configuration files—part of the centralized provisioning method—to provision and configure settings for multiple phones. Typically, settings you make using configuration files apply to multiple phones. Settings made using the Web Configuration Utility and the phone's user interface are applied on a per-phone basis and are available to individual phone users.

Tip: Administrative and User Settings

Settings available to administrators are not available to users and will not duplicate settings available to users. Be cautious about using multiple configuration methods for *administrative* settings.

Resetting to Default

There are five ways to reset or clear features and settings to the default values.

To reset the phone to the default values:

1. On the phone, go to **Settings > Advanced > Administration Settings > Reset to Defaults**.
2. Choose one of the following options:
   - **Reset Local Configuration** Clears the override file generated by changes using the phone user interface
   - **Reset Web Configuration** Clears the override file generated by changes using the Web Configuration Utility.
   - **Reset Device Settings** Resets the phone's flash file system settings that are not stored in an override file. These are your network and provisioning server settings and include custom certificates and encryption keys. Local, web, and other configuration files remain intact.

- **Format File System** Formats the phone's flash file system and deletes the UC software application, log files, configuration, and override files. Note that if the override file is stored on the provisioning server, the phone will re-download the override file when you provision the phone again. Formatting the phone's file system does not delete those device settings affecting network and provisioning, and any certificates and encryption keys remain on the phone.
- **Reset to Factory** Removes the web and local override files, any stored configuration files in the flash file system, as well as any custom certificates and encryption keys. All network and provisioning settings are reset but the UC software application and updater remain intact.

The rest of this section explains each of the following configuration methods:

- Use the Centralized Provisioning Method: Configuration Files
- Provision with the Web Configuration Utility
- Phone User Interface – Menu System Settings

Use the Centralized Provisioning Method: Configuration Files

Polycom recommends using a central provisioning server when your VoIP environment has multiple phones. Polycom provides template configuration files in XML format that you can use to create a set of phone features and settings specific to your organization. All of the phone features and settings are outlined in the following sections. The UC Software configuration files you use to configure the phones are very flexible. Parameters can be stored in the files in any order and can be placed in any number of files. You can change the XML tree structure, move parameters around within the XML files, change the file names, or create your own configuration files. These files dictate the behavior of the phone after it is running the Polycom UC Software. Be aware that the configuration files have default values that you may want to change.

Settings: Using the Default Value for a Configuration Parameter

The phone will use the default value for a configuration parameter as long as the parameter has not been configured from any other source. Parameters can be changed using the local phone user interface, the Web configuration utility, a Polycom CMA system, and configuration files hosted on a central provisioning server.

Applying configuration files to phones from a central provisioning server enables you to apply a single set of parameters and settings to all of the phones in your deployment. The configuration files maximize flexibility in installing the UC Software, configuring the phones, and in upgrading and maintaining the phone settings over time.

The CX5500 system can boot up without any configuration files; however, certain parameters need to be changed for your system to be usable within your organization. Note that if a system cannot locate a provisioning server upon boot up, the system operates with internally stored default settings. To send and receive calls, you must specify a SIP server address and a registration address (the equivalent of a phone number) in the configuration files.

You can create user-specific configuration files that enable phone users to use their features and settings from any phone including those outside of your organization. To create a user-specific file, create a

***<user>*.cfg** on the provisioning server for the user (including default user accounts). For more information, see Set User Profiles.

Settings: Choosing a Per-Phone Configuration File Name

Do not use <*MACaddress*>-phone.cfg, <*MACaddress*>-Web.cfg, <*MACaddress*>-app.log, or <*MACaddress*>-license.cfg as the per-phone filename—where the *MACaddress* is represented as a 12-digit number (for example, 000123456789). These filenames are used by the phone itself to store user preference overrides and logging information.

Understand the Master Configuration File

The centralized provisioning method requires you to use a master configuration file, named **00000000000.cfg** in the UC Software download. You can use the default master configuration file or you can create and rename a master configuration file to apply to phones in a network in one of the following ways:

- To all of the phones in a deployment
- To a group of phones in a deployment
- On a per-phone basis (to a single phone)
- In a specific location

Settings: Use the .cfg extension on the master configuration file.

The master configuration file must have the .cfg extension. No other configuration files must have the .cfg extension.

Each of these ways is described next in more detail.

- **Default master configuration file** For deployments in which the configuration is identical for all phones, you can use the default master configuration file, named **000000000000.cfg** in the UC Software download, to configure all the phones in a deployment. Note that the phones are programmed to look first for their own ***<MACaddress>*.cfg** file and if a phone does not find a matching file, it looks next for the default file. If you do create and use a per-phone master configuration file, make a copy of the default file and rename it.
- **Group and per-phone master configuration file** If you want to apply features or settings to a group of phones within your deployment or to a single phone, make a copy of the default file and rename it. For a phone group, rename the file in a way that specifies the group-specific features or settings. For single phones, rename the file based on the phone's MAC address ***<MACaddress>*.cfg**. The MAC address, also known as the serial number (SN), is a unique a-f hexadecimal digit assigned to each phone. Note that you can use only lower-case digits, for example, **0004f200106c.cfg**. You can find the MAC address of a phone on a label on the back of the phone or on the phone's menu system at **Settings > Status > Platform > Phone > S/N:** .

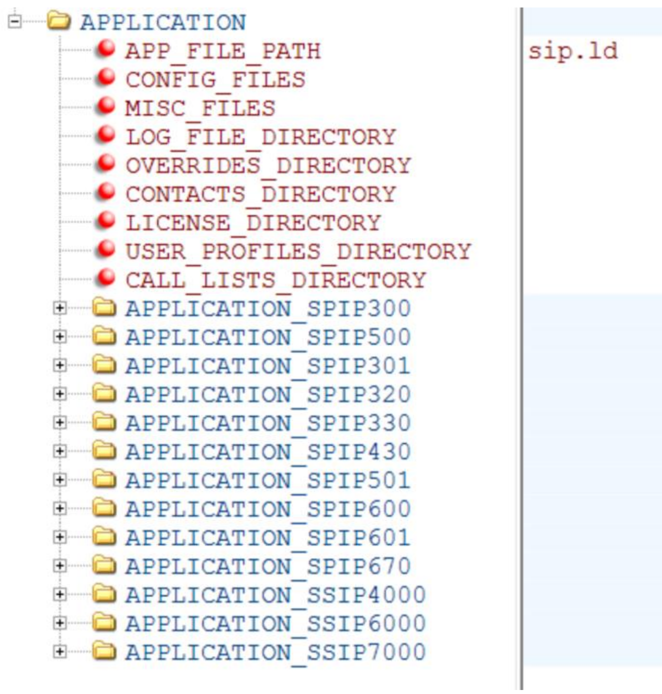
- **Specified master configuration file** You can specify a master configuration file in the provisioning server address, for example, http://usr:pwd@server/dir/example1.cfg. The filename must end with **.cfg** and be at least five characters long. If this file cannot be downloaded, the phone will search for a per-phone master configuration file, described next.

Settings: Pay Attention to Per-Phone File Names

Do not use the following names as extensions for per-phone files: <*MACaddress*>-phone.cfg, <*MACaddress*>-Web.cfg, <*MACaddress*>-app.log, , or <*MACaddress*>-license.cfg. These filenames are used by the phone to store override files and logging information.

The figure Default Fields in the Master Configuration File shows default available fields in the master configuration file, **000000000000.cfg**.

Default Fields in the Master Configuration File



The following describes each of the master configuration file XML attributes and the APPLICATION directories.

- **APP_FILE_PATH** Not applicable for CX5500 systems.
- **CONFIG_FILES** Enter the names of your configuration files here as a comma-separated list. Each file name has a maximum length of 255 characters and the entire list of file names has a maximum length of 2047 characters, including commas and white space. If you want to use a configuration file in a different location or use a different file name, or both, you can specify a URL with its own protocol, user name and password, for example, ftp://usr:pwd@server/dir/phone2034.cfg.

Settings: Order of the Configuration Files

The order of the configuration files listed in CONFIG_FILES is significant:

- The files are processed in the order listed (left to right).
- If the same parameter is included in more than one file or more than once in the same file, the first (left) parameter read is used.

- **MISC_FILES** A comma-separated list of files. You can use this to list volatile files that you want phones to download, for example, fonts, background images, or ringtone .wav files. The phone downloads files you list here when booted, which can decrease access time.
- **LOG_FILE_DIRECTORY** An alternative directory to use for log files if required. A URL can also be specified. This is blank by default.
- **CONTACTS_DIRECTORY** An alternative directory to use for user directory files if required. A URL can also be specified. This is blank by default.
- **OVERRIDES_DIRECTORY** An alternative directory to use for configuration overrides files if required. A URL can also be specified. This is blank by default.
- **LICENSE_DIRECTORY** An alternative directory to use for license files if required. A URL can also be specified. This is blank by default.
- **USER_PROFILES_DIRECTORY** An alternative directory for the **<*user*>.cfg** files.
- **CALL_LISTS_DIRECTORY** An alternative directory to use for user call lists if required. A URL can also be specified. This is blank by default.
- **COREFILE_DIRECTORY** An alternative location for phones that can upload a core file containing debugging with diagnostic when they fail. This is blank by default

The directories labeled **APPLICATION_SPIP*XXX*** indicate phone models that are not compatible with the latest UC Software version. If you are using any of the phone models listed in these directories, open the directory for the phone model you are deploying, and use the available fields to provision and configure those phones.

Understand Variable Substitution

The master configuration template file, included in the UC Software files you download from the Polycom Voice Support Web site, is particularly important to the central provisioning method, which Polycom recommends using for large-scale deployments. There are two methods you can use to provision or configure phones with the master configuration file. The method you use depends on your deployment scenario. Understanding both methods enables you to deploy and manage your phones efficiently. For a detailed explanation of the two methods and their advantages, see *Best Practices 35361: Provisioning with the Master Configuration File*.

You can also use variable substitution if you need to use different application loads on different phones on the same provisioning server by creating a variable in the master configuration file that is replaced by the MAC address of each phone when it reboots. You can use any of the following substitution strings:

- PHONE_MODEL
- PHONE_PART_NUMBER
- PHONE_MAC_ADDRESS

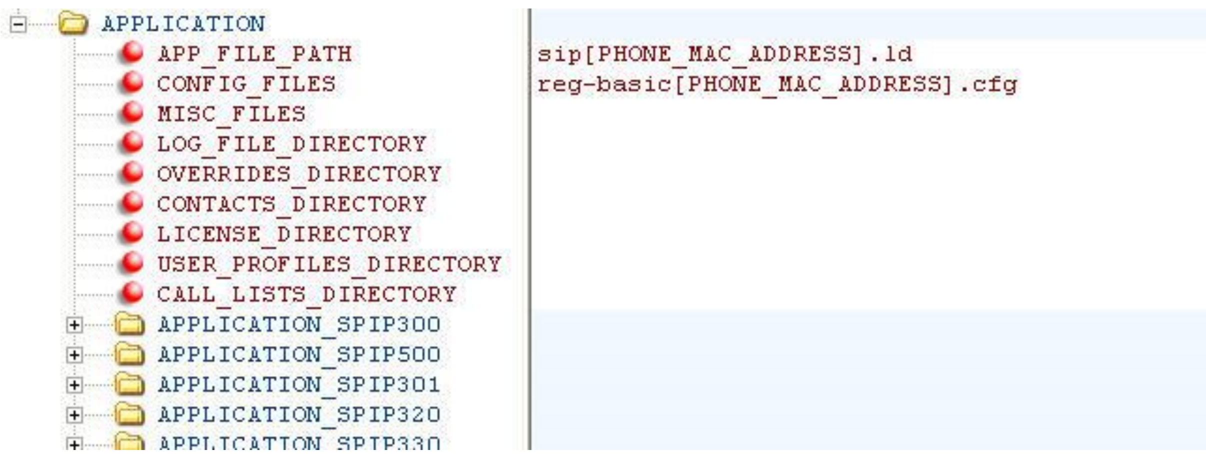
To find out the model number or part number of a product, see the section Product, Model, and Part Number Mapping.

The following two examples illustrate the use of a variable substitution.

Example One

You can create a variable in the master configuration file that is replaced by the MAC address of each phone when it reboots as shown in the figure MAC Address Variable.

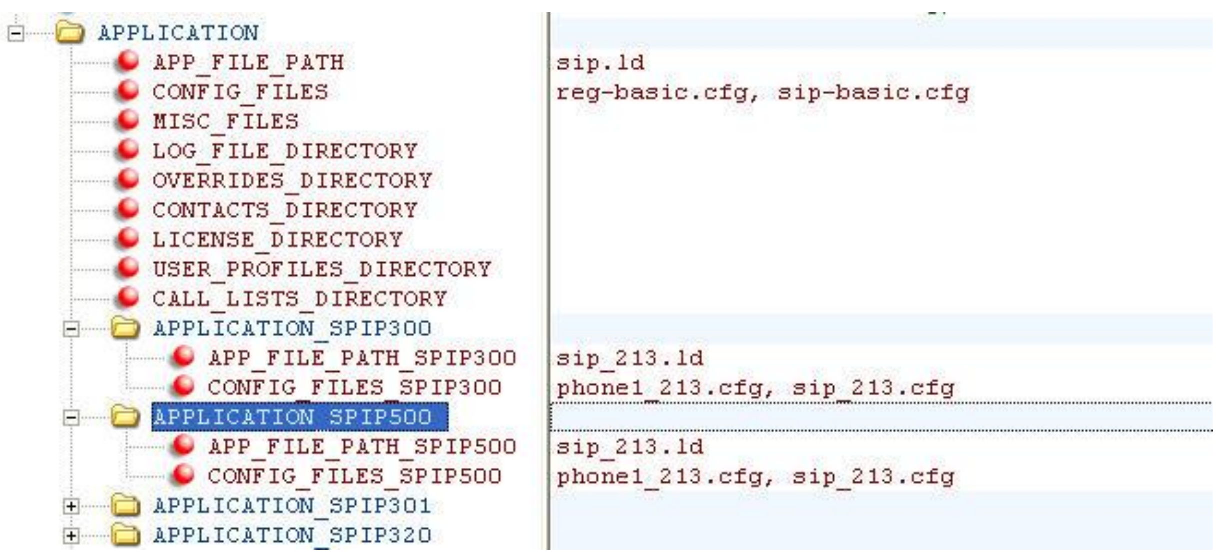
MAC Address Variable



Example Two

You can direct phone update to a UC software build and configuration files based on the phone model number and part number as shown in the figure Provisioning with Model and Part Numbers. All XML attributes can be modified in this manner.

Provisioning with Model and Part Numbers



Use the Template Configuration Files

You will find a number of template configuration files when you expand the Polycom CX5500 software download. Most configuration parameters are located in only one template file; however, some do appear in two or more files. If you are using a parameter that is duplicated in another file, be aware that configuration files are read from left to right and the phone uses the file it reads first.

Troubleshooting: Locating Duplicate Parameters

To check whether a parameter is located in more than one template file, locate the parameter in the reference section Configuration Parameters.

The table Configuration File Templates shown next outlines each template file included with the CX5500 software.

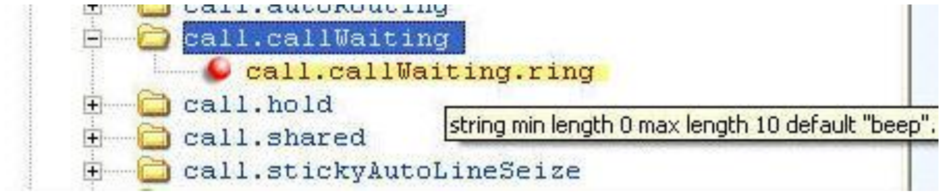
Configuration File Templates

Name	*Description*	*Deployment Scenarios*
applications.cfg	For applications, browser, microbrowser, XMP-API	Typical Hosted Service Provider Typical IP-PBX
features.cfg	Features related enabling corp directory USB recording, CMA, presence, ACD, for example	Typical Hosted Service Provider Typical IP-PBX
reg-advanced.cfg	Advanced call server, multi-line phones	Typical Hosted Service Provider Typical IP-PBX
reg-basic.cfg	Basic registration	Simple SIP device Typical Hosted Service Provider
region.cfg	Non-North American geographies	Typical Hosted Service Provider Typical IP-PBX
sip-basic.cfg	Basic call server	Simple SIP device Typical Hosted Service Provider
sip-interop.cfg	Advanced call server, multi-line phones	Typical Hosted Service Provider Typical IP-PBX
site.cfg	Multi-site operations	Typical Hosted Service Provider Typical IP-PBX
techsupport.cfg	Available by special request from Polycom Customer Support.	Troubleshooting

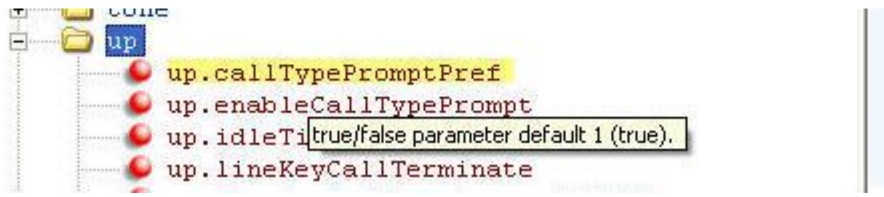
Along with the templates, the CX5500 software download includes an XML schema file—polycomConfig.xsd—that provides information like parameters type (boolean, integer, string, and enumerated type), permitted values, default values, and all valid enumerated type values. View this template file with an XML editor.

A string parameter, boolean, and enumerated parameters are shown in the following figures String Parameter, Boolean Parameter, and Enumerated Parameter.

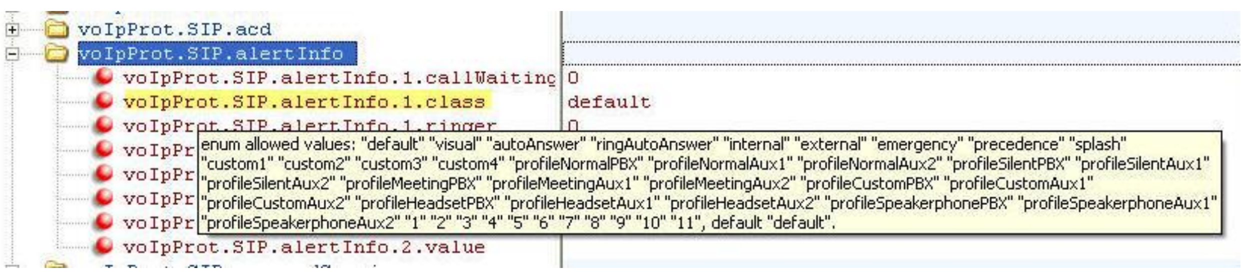
String Parameter



Boolean Parameter



Enumerated Parameter



Change Configuration Parameter Values

The configuration parameters available in the UC Software use a variety of values, including Boolean, integer, enumerated types, and arrays (a table of values). Each parameter available in the UC Software is listed in alphabetical order in Configuration Parameters, along with a description, the default value, and the permissible values.

Note that the values for boolean configuration parameters are not case sensitive. The values `0`, `false`, and `off` are inter-changeable and supported. The values `1`, `true`, and `on` are interchangeable and supported. This Administrator’s Guide documents only `0` and `1`.

The following rules apply when you set a parameter with a numeric value outside of its valid range:

- If the configuration file’s value is greater than the allowable range, the maximum value is used.
- If the configuration file’s value is less than the allowable range, the minimum value is used.

- If a parameter's value is invalid, the value is ignored. Invalid parameters values can occur when enumerated type parameters do not match a pre-defined value, when numeric parameters are set to a non-numeric values, when string parameters are either too long or short, or when using null strings in numeric fields. All such situations are logged in the phone's log files.

Tip: Using Blank Values and Special Characters in the Configuration Files

The UC Software interprets Null as empty; that is, attributeName="".

To enter special characters in a configuration file, enter the appropriate sequence using an XML editor:

- & as &
- " as "
- ' as '
- < as <
- > as >
- random numbers as &0x12;

Customize Parameters for a Phone Model

You can customize a set of parameter values for the CX5500 system by appending the PHONE MODEL NUMBER descriptor to the parameter. For a list of all phone model names that you can use to create phone-specific configurations, see Product, Model, and Part Number Mapping.

For example:

- dir.local.contacts.maxNum="9999"
- dir.local.contacts.maxNum.CX5500="500"

In this example, the maximum number of contacts for the local Contact Directory on the CX5500 system is 500.

Some configuration parameters cause the phone to reboot or restart when change its value. To find out if a parameter reboots or restarts a phone when changed, locate the parameter in Configuration Parameters. Parameters that reboot or restart the phone are marked with a superscript (^{1}or ^{2}).

Caution: Deprecated Configuration Parameters

Polycom may deprecate configuration parameters that some organizations may still be using—deprecated parameters will not work. To check whether or not you are using deprecated configuration parameters, see the latest Polycom UC Software Release Notes on Latest Polycom UC Software Release or check the Release Notes for earlier software versions on Polycom UC Software Support Center.

Provision with the Web Configuration Utility

The Web Configuration Utility enables you to perform configuration changes on a per-phone basis. If you are provisioning more than ten or twenty phones, Polycom recommends using centralized provisioning as your primary configuration method.

Admin Tip: Updating UC Software on a Single Phone

You can use the Software Upgrade tool in the Web Configuration Utility to update the UC Software version running on a single phone. For detailed information, see *Feature Profile 67993: Using the Software Upgrade Tool in the Web Configuration Utility*.

Web Info: Using the Web Configuration Utility

For more detailed help navigating and using the Web Configuration Utility, see the *Polycom Web Configuration Utility User Guide*.

You can access the Web Configuration Utility using any of the following Web browsers:

- Microsoft® Internet Explorer 7.0 or later
- Mozilla® Firefox® 3.0.X or later
- Google Chrome™ 10.0.X or later
- Apple® Safari® 5.0.4 or later

The Web Configuration Utility comes with built-in contextual help functions that provide you with information and guidance on how to perform basic phone configuration changes. In addition, you can choose to display the interface of the Web Configuration Utility in one of several languages.

Access the Web Configuration Utility

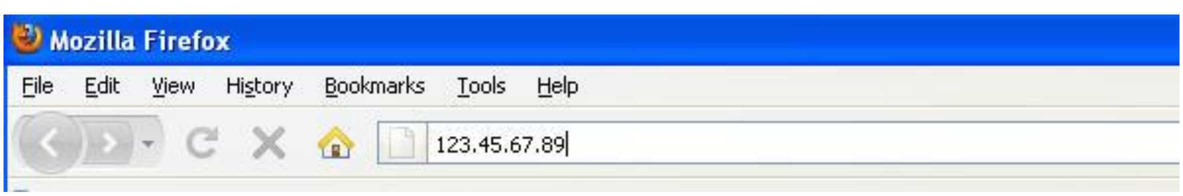
You can access the Web Configuration Utility by entering the phone's IP address in a supported Web browser, for example, http://<*phone IP address*>. If you are a user, log in as **User**; the default password is **123**. If you are an administrator, log in as **Admin**; the default password is **456**.

To access the Web Configuration Utility:

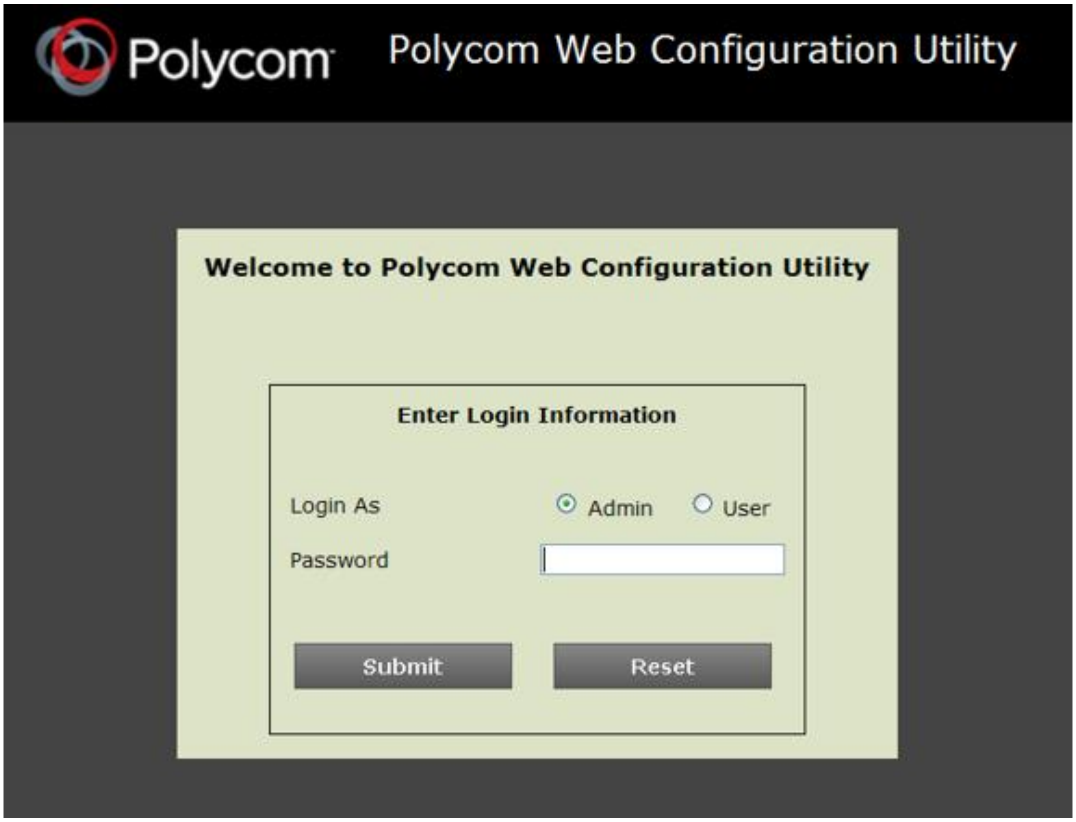
1 Open one of the supported Web browsers.

Get your phone's IP address. On the phone, select **Settings > Status > Platform > Phone**. Scroll down to see the IP address.

2 Enter your phone's IP address in the browser's address bar.

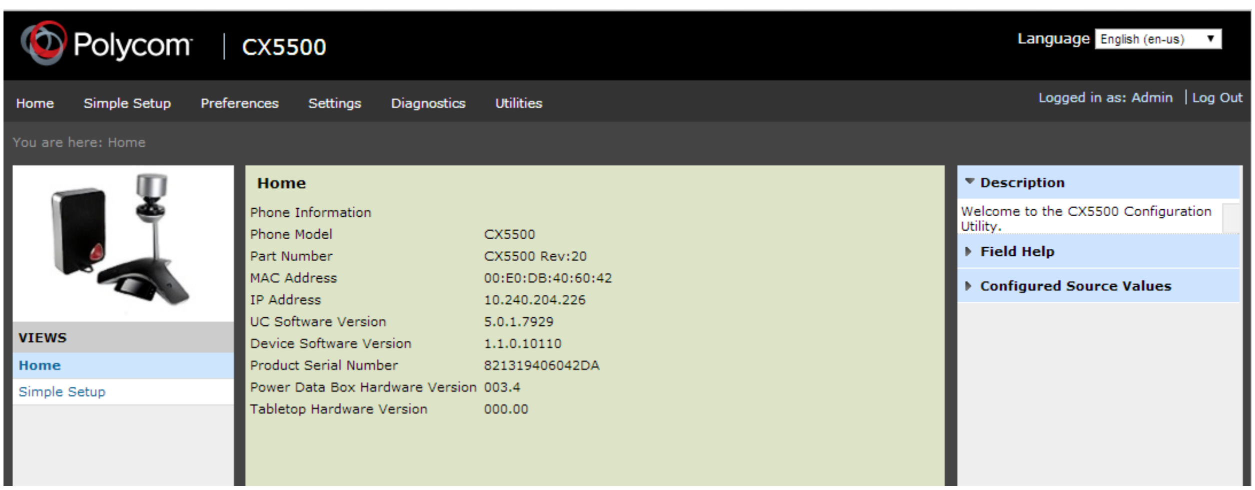


A web page similar to the one shown next displays.



3 Log in as **Admin**—the default administrative password is **456**.

A web page similar to the one shown next displays.



4 Use the menus to navigate through the available settings. The sidebar on the right gives you description of the each page, contextual field help, and the parameter for each setting.

Choose Language Files for the Web Configuration Utility Interface

You can choose a language for viewing the Web Configuration Utility interface. Polycom provides a number of XML language files that you can download from the Polycom CX5500 software package to your provisioning server. By default, the phone displays the Web Configuration Utility in English. If you want the phone to display the Web Utility interface in a language other than English, copy the corresponding XML language file from the languages folder to your provisioning server. This section shows you how to copy the Web Configuration Utility language files to your provisioning server so that phone users can use the Web Configuration Utility interface in the language of their choice.

Certain languages available on the CX5500 system use an expanded character set and more memory than other language files. On average, the XML language files for the Web Configuration Utility interface are about 250KB in size. To conserve memory resources, Polycom recommends using only those XML language files for the languages you need. If you want to make multiple languages available to your users, you may need to manage the phone's memory resources. For tips on how to do this, see Manage the Phone's Memory Resources.

To save XML language files to your provisioning server:

1 Create a new folder named *languages* on your provisioning server. This is the folder the provisioning server reads to apply language files to the interface of the Web Configuration Utility. For help setting up your provisioning server, see Set Up the Provisioning Server.

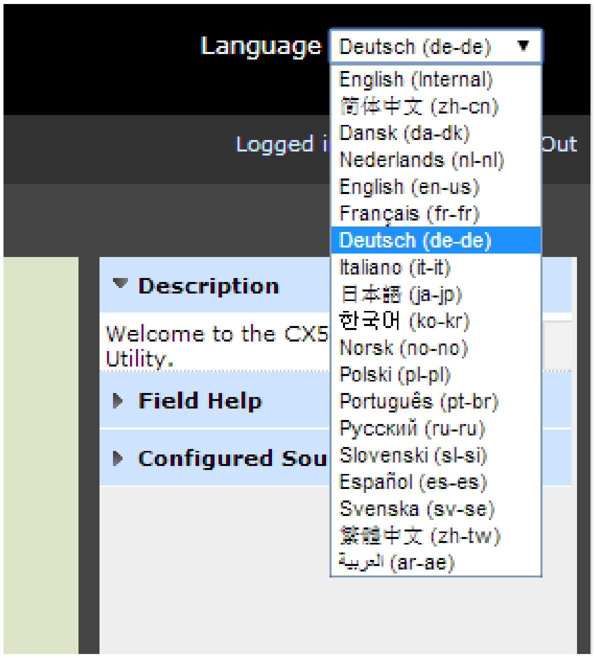
2 Download and unzip the UC software package. You can find all of the language files for the Web Configuration Utility interface in a folder named *languages*.

Note: Don't Confuse Language Files

The *languages* folder located in both the combined and split UC Software versions is not to be confused with the *language files* for the phone interface, which are located in the SoundPointIPLocalization folder. To save memory on the phone, Polycom recommends that you save only the Web Configuration Utility language files that you need to the languages folder you created in your provisioning server.

3 Copy the XML language file from the languages folder you downloaded from the software files to the languages folder you created on your provisioning server. For example, if you want the Web Configuration Utility to support French and German, copy `Website_dictionary_language_fr-fr.xml` and `Website_dictionary_language_de-de.xml` to the languages folder you created on your provisioning server.

4 Log in to the Web Configuration Utility and select a language from the **Languages** drop-down menu at the top-right of the screen, as shown next.



The interface of the Web Configuration Utility displays in the language you select. If the language does not display, ensure that you have extracted and saved the correct language file, or try rebooting the phone.

Troubleshooting: Managing the Phone’s Memory Resources

If your selected language will not display, even after you have placed it on the provisioning server and you have rebooted the phone, your phone may have reached its available memory limit. If this occurs, you may need to take steps to manage your phone’s available memory resources. For tips on how to manage the phone’s memory, refer to Manage the Phone’s Memory Resources.

Phone User Interface – Menu System Settings

The phone menu system makes some settings available to users and further settings available to administrators.

To access administrator settings, such as provisioning values, enter an administrative password in the **Settings** menu on the LCD panel. Note that you can use an administrator password where a user password is required, but a user cannot access administrator settings with a user password. The default user password is **123** and the default administrative password is **456**. To secure the administrative settings from the phone’s user interface, change the default administrative password. See Local User and Administrator Passwords.

Timesaver: Phone User Interface Menu System

For a map diagram of all menu settings available from the phone user interface, see Polycom UC Software Menu System.

Set Up Basic Phone Features

After you set up your Polycom® phones with a default configuration on the network, phone users will be able to place and receive calls. However, you may want to add features to the default configuration to suit your organization and user's needs. Polycom provides basic and advanced features that you can configure for the phones to add efficiency and convenience. This section will show you how to configure all available basic phone features and call management features.

Before you begin configuring phone features, take the time to read the short introductory section Read the Feature Parameter Tables. This section provides important information you need to know in order to successfully perform configuration changes.

Basic Phone Features at a Glance

This section shows you how to make configuration changes for the following basic features:

- Configure the Call Logs   Contains call information such as remote party identification, time and date, and call duration in three separate lists, missed calls, received calls, and placed calls.
- Understand the Call Timer   Maintains a timer, in hours, minutes, and seconds, for each call in progress.
- Configure Call Waiting Alerts   Visually presents an incoming call on the screen, and plays a configurable sound effect, when you're in another call.
- Called Party Identification   Displays and logs the identity of the party in an outgoing call.
- Configure Calling Party Identification   Displays a caller's identity, derived from the network signaling, when an incoming call is presented—if the information is provided by the call server.
- Connected Party Identification   Displays and logs the identity of the party to whom you are connected to (if the name is provided by the call server).
- Distinctive Incoming Call Treatment   Automatically applies distinctive treatment to calls containing specific attributes.
- Apply Distinctive Ringing   Enables you to select a ring tone for each line, as well as a ring tone for contacts in the contact directory.
- Apply Distinctive Call Waiting   Enables you to map calls to distinct call waiting types.
- Configure Do Not Disturb   Temporarily stops incoming calls.
- Use the Local Contact Directory   The phone maintains a local contact directory that can be downloaded from the provisioning server and edited locally. Any edits to the Contact Directory made on the phone are saved to the provisioning server as a backup.
- Configure the Local Digit Map   The phone has a local set of rules to automate the setup phase of number-only calls.
- Microphone Mute   Mutes the phone's microphone so other parties cannot hear you. When the microphone mute feature is activated, the mute buttons on the system glow red.
- Configure the Speed Dial Feature   Enables you to place calls quickly from dedicated keys as well as from a speed dial menu.

- Set the Time and Date Display   Time and date can be displayed in certain operating modes such as when the phone is idle and during a call.
- Set a Graphic Display Background   Enables you to display a picture or graphic on the screen's background.

This section also shows you how to make configuration changes for the following basic call management features:

- Enable Automatic Off-Hook Call Placement   Supports an optional automatic off-hook call placement feature for each registration.
- Configure Call Hold   Pauses activity on one call so that you can use the phone for another task, such as making or receiving another call.
- Use Call Transfer   Transfers a call in progress to some other destination.
- Create Local and Centralized Conferences   You can host or join local conferences or create centralized conferences using conference bridge numbers. The advanced aspects of conferencing, like managing parties, are part of the Productivity Suite.
- Enable Conference Management   Add, hold, mute, and remove conference participants, and obtain information about participants.
- Configure Call Forwarding   Provides a flexible call forwarding feature to forward calls to another destination.
- Configure Lync Call Forwarding   Provides a flexible call forwarding feature for CX5500 systems registered with Microsoft Lync Server.
- Configure Directed Call Pick-Up and Enable Group Call Pickup   Enables you to pick up calls to another phone by dialing the extension of the other phone. Calls to another phone within a pre-defined group can be picked up without dialing the extension of the other phone.
- Configure Call Park and Retrieve   Park an active call—puts it on hold to a specific location, so it can be retrieved by any phone.
- Enable Last Call Return   Automatically redials the number of the last received call.

Configure the Call Logs

The phone records and maintains phone events to a call log, also known as a call list. These call logs contain call information such as remote party identification, time and date of the call, and call duration. The log is stored as a file in XML format named **<*MACaddress*>-calls.xml** to your provisioning server. If you want to route the call logs to another server, use the CALL_LISTS_DIRECTORY field in the master configuration file. You can use the call logs to redial previous outgoing calls, return incoming calls, and save contact information from call log entries to the contact directory. All call logs are enabled by default. See the table Configure the Call Logs for instructions on how to enable or disable the call logs.

The phones automatically maintain the call logs in three separate call lists: Missed Calls, Received Calls, and Placed Calls. Each of these call lists can be cleared manually by individual phone users. You can delete individual records or all records in a group (for example, all missed calls). You can also sort the records or filter them by line registration.

Tip: Merged Call Lists

On some phones, missed and received calls display in one call list. In these combined lists, you can identify call types by the icons:

- Missed call icon 
- Received call icon 

Configure the Call Logs

Central Provisioning Server	template > parameter
Enable or disable the missed call list	**features.cfg** > feature.callListMissed.enabled
Enable or disable the placed call list.	**features.cfg** > feature.callListPlaced.enabled
Enable or disable the received call list	**features.cfg** > feature.callListReceived.enabled

Example Call Log Configuration

The following illustration shows you each of the call log parameters you can enable or disable in the **features.cfg** template file.

The table Call Log Elements and Attributes describes each element and attribute that displays in the call log. Polycom recommends using an XML editor such as XML Notepad 2007 to view and edit the call log. Note that you can place the elements and attributes in any order in your configuration file.

Call Log Elements and Attributes

Element	*Permitted Values*
direction	**In, Out**
Call direction with respect to the user.	
disposition	**Busy, Forwarded, Normal, Partial, Preempted, Rejected, RemotelyHandled, Transferred**
What happened to the call. When a call entry is first created, the disposition is set to Partial.	
line	**Positive integer**
The line (or registration) index.	
protocol	**SIP or H323**
The line protocol.	
startTime	**String**
The start time of the call. For example: 2010-01-05T12:38:05 in local time.	
duration	**String**
The duration of the call, beginning when it is connected and ending when the call is terminated. For example: `PT1H10M59S`.	
count	**Positive Integer**
The number of consecutive missed and abandoned calls from a call destination.	
destination	**Address**
The original destination of the call. For outgoing calls, this parameter designates the outgoing call destination; the name is initially supplied by the local phone (from the name field of a local contact entry) but may later be updated via call signaling. This field should be used for basic redial scenarios. For incoming calls, the called destination identifies the requested party, which may be different than any of the parties that are eventually connected (the destination may indicate a SIP URI which is different from any SIP URI assigned to any lines on the phone).	
source	**Address**
The source of the call (caller ID from the call recipient's perspective).	
Connection	**Address**
An array of connected parties in chronological order. As a call progresses, the connected party at the far end may change, for example, if the far end transfers the call to someone else. The connected element allows the progression of connected parties, when known, to be saved for later use. All calls that contain a connected state must have at least one connection element created.	
finalDestination	**Address**
The final connected party of a call that has been forwarded or transferred to a third party.	

Understand the Call Timer

A call timer displays on the phone's screen. A separate call duration timer displays the hours, minutes, and seconds of each call in progress.

There are no related configuration changes.

Configure Call Waiting Alerts

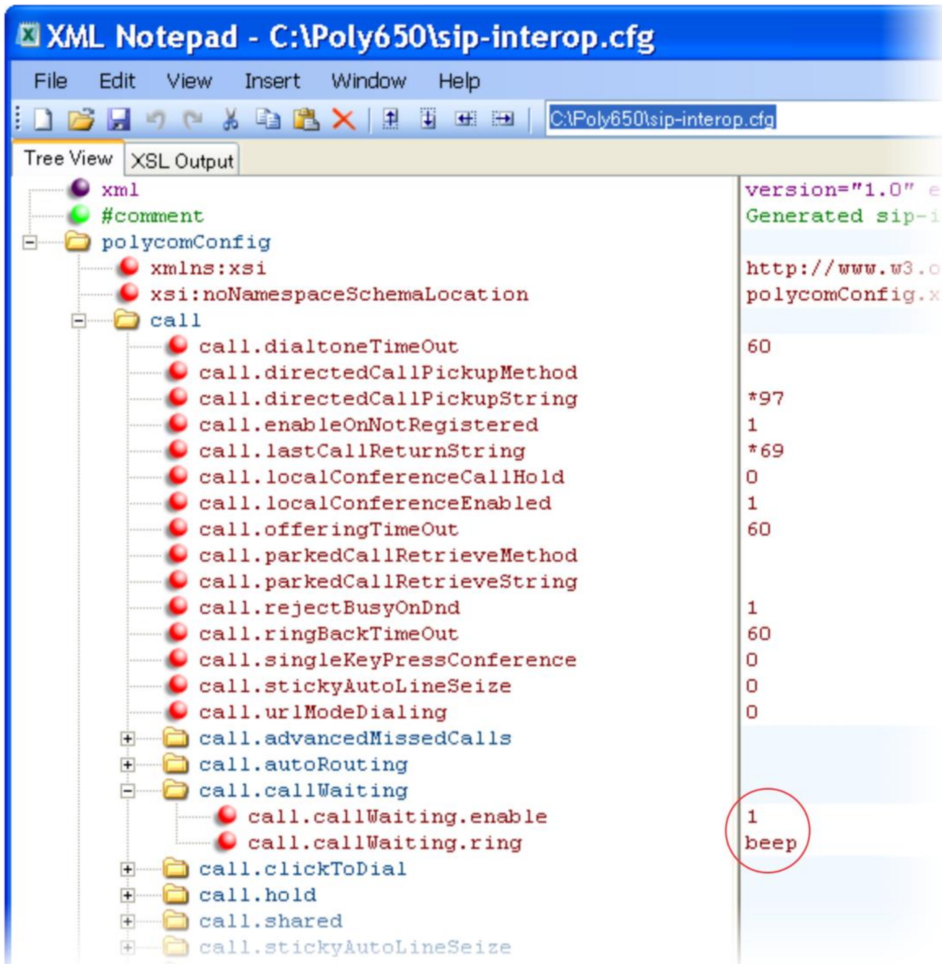
By default, the phone will alert you to incoming calls while you are in an active call. As shown in the table Configuring Call Waiting Alerts, you can disable call waiting alerts and you can specify the ringtone of incoming calls.

Configuring Call Waiting Alerts

Central Provisioning Server	**template** > parameter
Enable or disable call waiting	**sip-interop.cfg** > call.callWaiting.enable
Specify the ringtone of incoming calls when you are in an active call	**sip-interop.cfg** > call.callWaiting.ring

Example Call Waiting Configuration

The following illustration shows you where to disable call waiting alerts and how to change the ringtone of incoming calls in the **sip-interop.cfg** template.



Called Party Identification

By default, the phone displays and logs the identity of parties called from the phone. The phone obtains called party identity from the network signaling. Because Called Party Identification is a default state, the phone will display caller IDs matched to the call server and does not match IDs to entries in the Local Contact Directory or Corporate Directory.

There are no related configuration changes.

Configure Calling Party Identification

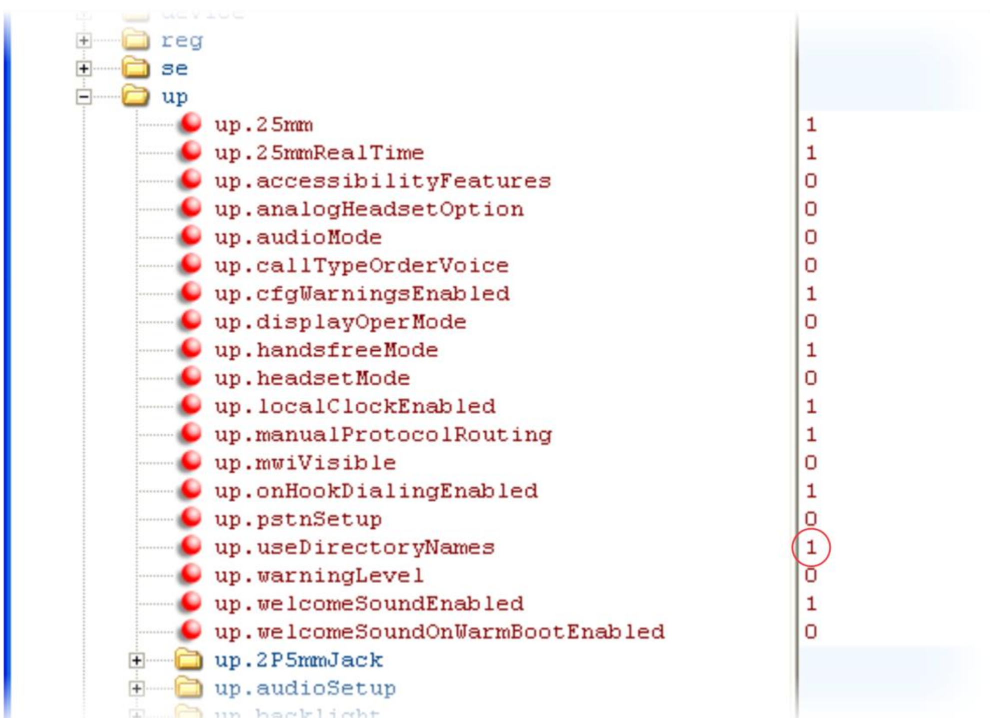
By default, the phone displays the identity of incoming callers if available to the phone through the network signal. If the incoming call address has been assigned to the contact directory, you can choose to display the name you assigned there, as shown in the following table. Note that the phone cannot match the identity of calling parties to entries in the Corporate Directory.

Configuring Calling Party Identification

Central Provisioning Server	**template** > parameter
Substitute the network address ID with the Contact Directory name	**reg-advanced.cfg** > up.useDirectoryNames
Override the default number of calls per line key for a specific line	**reg-advanced.cfg** > reg.x.callsPerLineKey
Web Configuration Utility	
Specify whether or not to substitute the network address with the Contact Directory name. Navigate to **Preferences > Additional Preferences > User Preferences**.	

Example Calling Party Configuration

The following illustration shows you how to substitute the network address caller ID with the name you assigned to that contact in the contact directory. The ID of incoming call parties will display on the phone screen.



Enable Missed Call Notification

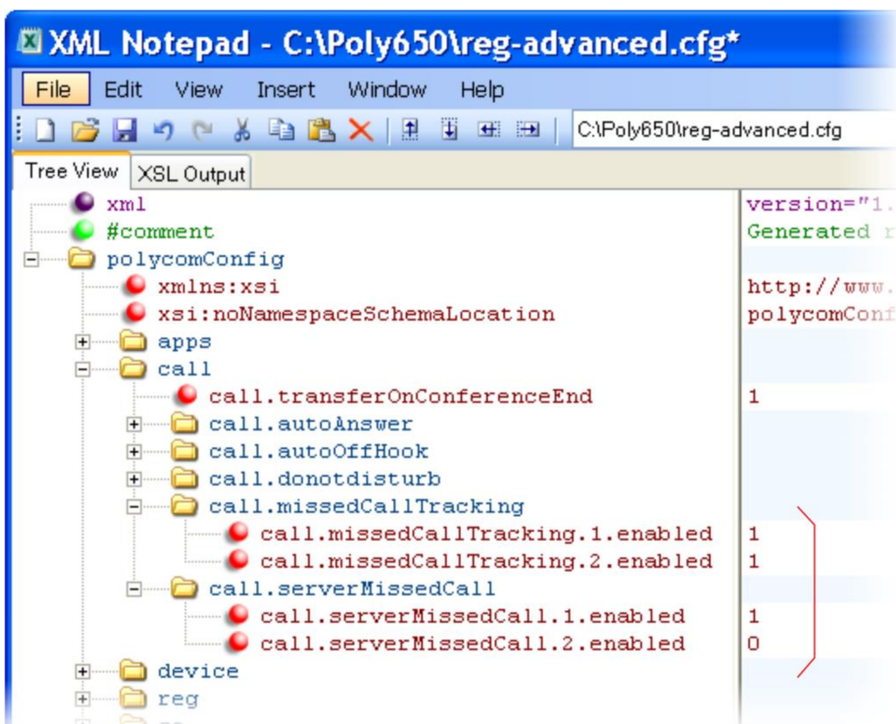
You can display on the phone's screen a counter that shows the number of missed calls. To reset the counter, view the Missed Calls list on the phone. As the table Enabling Missed Call Notification indicates, you can also configure the phone to record all missed calls or to display only missed calls that arrive through the Session Initiation Protocol (SIP) server. You can enable Missed Call Notification for each registered line on a phone.

Enabling Missed Call Notification

Central Provisioning Server	template > parameter
Enable or disable the missed call counter for a specific registration	**reg-advanced.cfg** > call.missedCallTracking.x.enabled
Specify, on a per-registration basis, whether to display all missed calls or only server-generated missed calls	**reg-advanced.cfg** > call.serverMissedCall.x.enabled

Example Missed Call Notification Configuration

In the following example, the missed call counter is enabled by default for registered lines 1 and 2, and only server-generated missed calls will be displayed on line 1.



Connected Party Identification

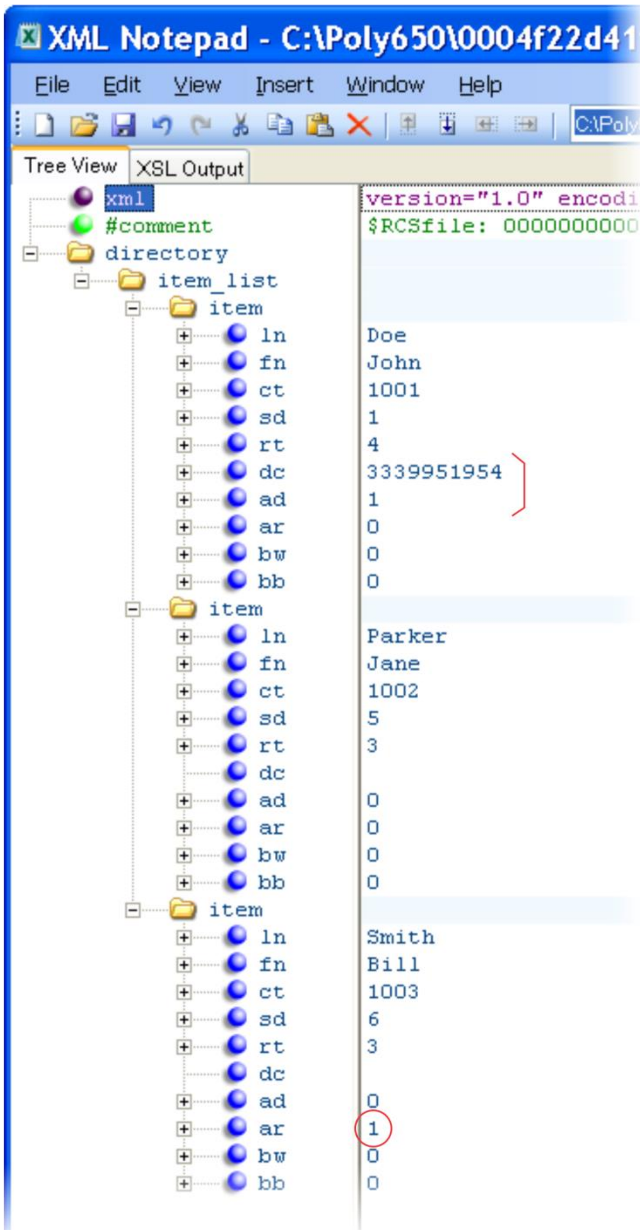
By default, the phone displays and logs the identity of remote parties you connect to if the call server can derive the name and ID from the network signaling. Note that in cases where remote parties have set up certain call features, the remote party you connect to—and the caller ID that displays on the phone—may be different than the intended party. For example, Bob places a call to Alice, but Alice has call diversion configured to divert Bob's incoming calls to Fred. In this case, the phone will log and display the connection between Bob and Fred. Note that the phone does not match party IDs to entries in the contact directory or the corporate directory.

Distinctive Incoming Call Treatment

You can apply distinctive treatment to specific calls and contacts in your contact directory. You can set up distinctive treatment for each of your contacts by specifying a *Divert Contact*, enabling *Auto-Reject*, or by enabling *Auto-Divert* for a specific contact in the local contact directory (see Use the Local Contact Directory. You can also apply distinctive treatment to calls and contacts through the phone's user interface.

Example Call Treatment Configuration

In the following example, the Auto Divert feature has been enabled in `ad` so that incoming calls from John Doe will be diverted to SIP address *3339951954* as specified in `dc`. Incoming calls from Bill Smith have been set to Auto Reject in `ar` and will be sent to voicemail.



Note that if you enable both the Auto Divert and Auto Reject features, Auto Divert has precedence over Auto Reject. For a list of all parameters you can use in the contact directory, see the table Understanding the Local Contact Directory.

Apply Distinctive Ringing

The distinctive ringing feature enables you to apply a distinctive ringtone to a registered line, a specific contact, or type of call.

There are three ways to set distinctive ringing, and the table Apply Distinctive Ringing shows you the parameters for each. If you set up distinctive ringing using more than one of the following methods, the phone will use the highest priority method.

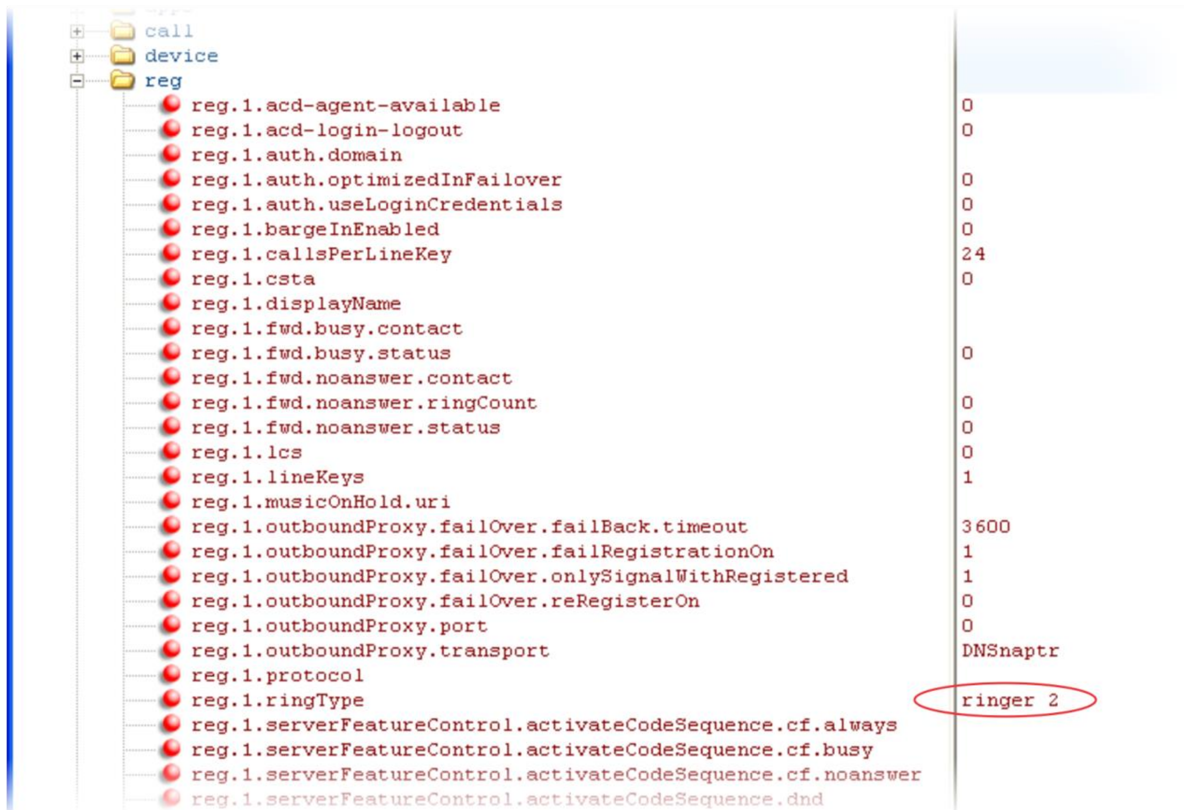
- You can assign ringtones to specific contacts in the Contact Directory. For more information, see Distinctive Incoming Call Treatment. This option is first and highest in priority.
- You can use the `voIpProt.SIP.alertInfo.x.value` and `voIpProt.SIP.alertInfo.x.class` parameters in the sip-interop.cfg template to map calls to specific ringtones. The value you enter depends on the call server. This option requires server support and is second in priority.
- You can select a ringtone for each registered line on the phone. Select **Settings > Basic > Ring Type**. This option has the lowest priority.

Apply Distinctive Ringing

Central Provisioning Server	**template** > parameter
Map alert info string in the SIP header to ringtones	**sip-interop.cfg** > voIpProt.SIP.alertInfo.x.class **sip-interop.cfg** > voIpProt.SIP.alertInfo.x.value
Specify a ringtone for a specific registered line	**reg-advanced.cfg** > reg.x.ringType
Specify ringtones for contact directory entries	**000000000000-directory~.xml**
Local Phone User Interface	
You can edit the ringtone of each registered line by navigating to **Settings > Basic > Ring Type**. To edit the ringtone for a specific contact, navigate **to Settings > Features > Contact Directory**, highlight a contact, tap the **Edit** soft key, and specify a value for the **Ring Type**.	

Example Distinctive Ringing Configuration

The following illustration shows that the ring type `ringer2` has been applied to incoming calls to line 1.



For a list of all parameters and their corresponding ringtones, see Ringtone Pattern Names.

Apply Distinctive Call Waiting

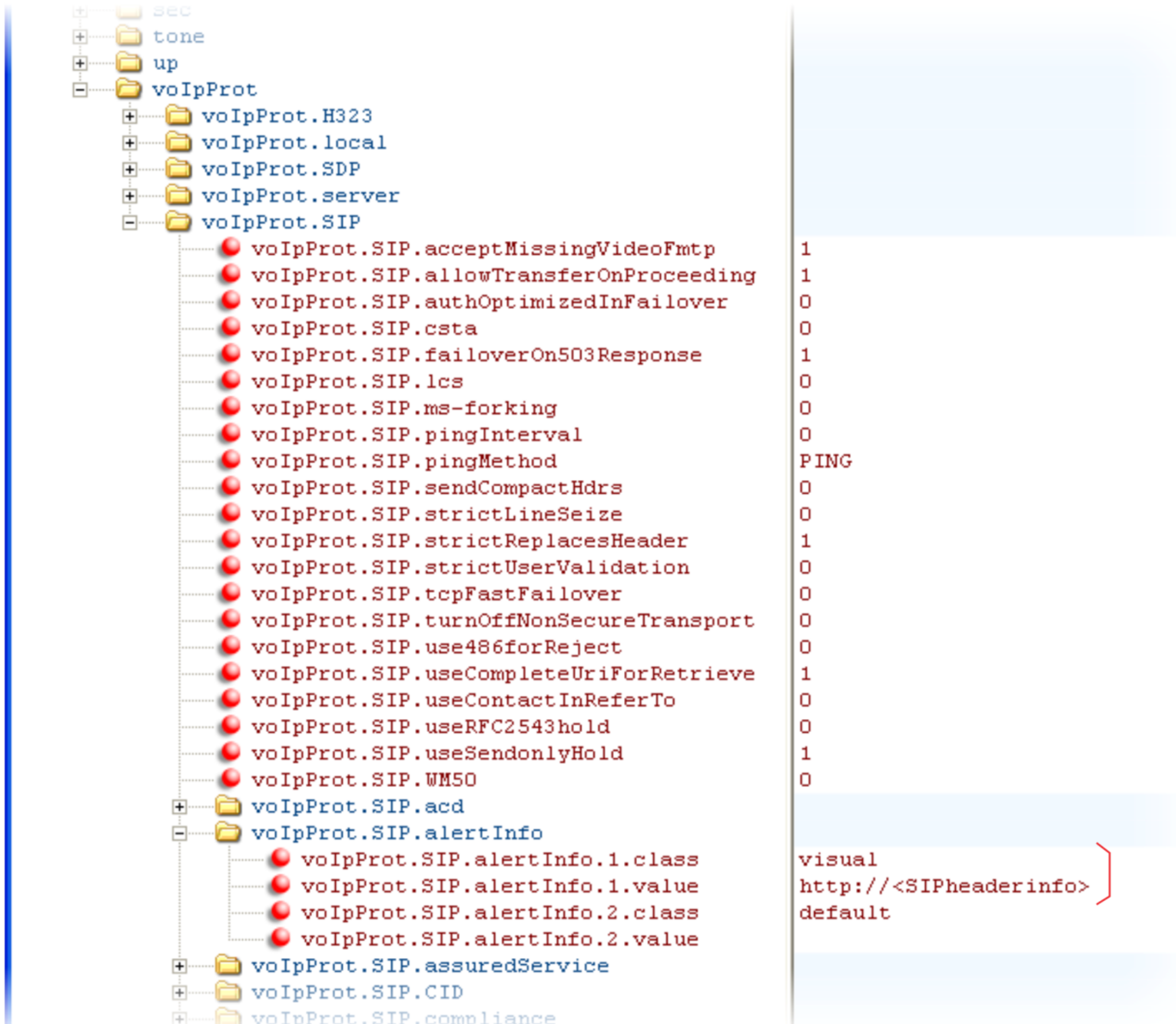
You can use the alert-info values and class fields in the SIP header to map calls to distinct call-waiting types. You can apply three call waiting types: beep, ring, and silent. The table Apply Distinctive Call Waiting shows you the parameters you can configure for this feature. This feature requires call server support.

Apply Distinctive Call Waiting

Central Provisioning Server	template > parameter
Enter the string which displays in the SIP alert-info header	**sip-interop.cg** > voIpProt.SIP.alertInfo.x.value
Enter the ring class name	**sip-interop.cfg** > voIpProt.SIP.alertInfo.x.class

Example Distinctive Call Waiting Configuration

In the following illustration, `voIpProt.SIP.alertInfo.1.value` is set to *http://<SIP headerinfo>*. An incoming call with this value in the SIP alert-info header will cause the phone to ring in a manner specified by `voIpProt.SIP.alertInfo.x.class`. In this example, the phone will display a visual LED notification, as specified by the value `visual`.



Configure Do Not Disturb

You can use the Do Not Disturb (DND) feature to temporarily stop incoming calls. You can also turn off audio alerts and receive visual call alerts only, or you can make your phone appear busy to incoming callers. Incoming calls received while DND is turned on are logged as missed.

DND can be enabled locally through the phone or through a server. The table Configure Do Not Disturb lists parameters for both methods. The local DND feature is enabled by default, and you have the option of disabling it. When local DND is enabled, you can turn DND on and off using the **Do Not Disturb** button on the phone. Local DND can be configured only on a per-registration basis. If you want to forward calls while DND is enabled, see Configure Call Forwarding.

Note: Using Do Not Disturb on Shared Lines

A phone that has DND enabled and activated on a shared line will visually alert you to an incoming call, but the phone will not ring.

If you want to enable server-based DND, you must enable the feature on both a registered phone and on the server. The benefit of server-based DND is that if a phone has multiple registered lines, you can apply DND to all line registrations on the phone; however, you cannot apply DND to individual registrations on a phone that has multiple registered lines. Note that although server-based DND disables the local Call Forward and DND features, if an incoming is not routed through the server, you will still receive an audio alert.

Server-based DND behaves the same way as the pre-SIP 2.1 per-registration feature with the following exceptions:

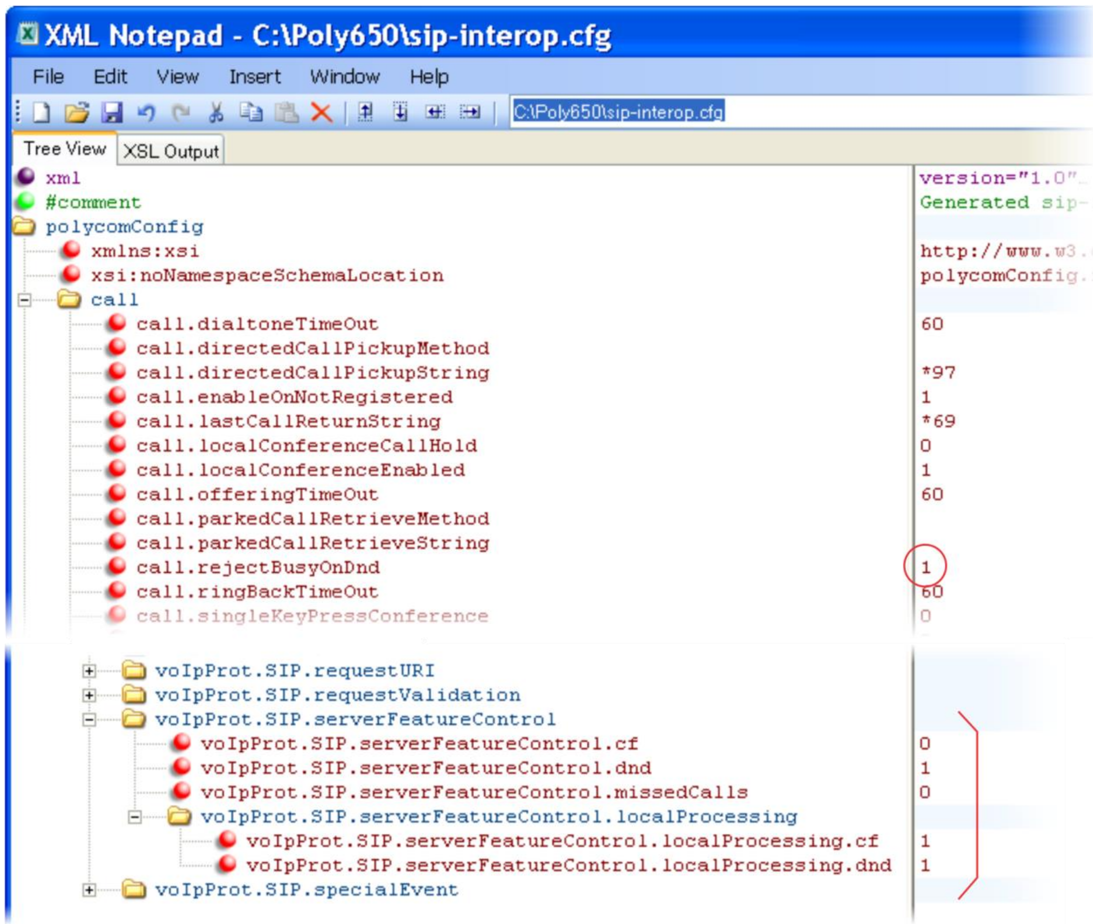
- You cannot enable server-based DND if the phone is configured as a shared line.
- If server-based DND is enabled but not turned on, and you press the DND key or select DND on the phone's Features menu, the "Do Not Disturb" message will display on the phone and incoming calls will continue to ring.

Configure Do Not Disturb

Central Provisioning Server	**template** > parameter
Enable or disable server-based DND	**sip-interop.cfg** > voIpProt.SIP.serverFeatureControl.dnd
Enable or disable local DND behavior when server-based enabled	**sip-interop.cfg** > voIpProt.SIP.serverFeatureControl.localProcessing.dnd
Specify whether, when DND is turned on, the phone rejects incoming calls with a busy signal or gives you a visual and no audio alert.	**sip-interop.cfg** > call.rejectBusyOnDnd
Enable DND as a per-registration feature or use it as a global feature for all registrations	**reg-advanced.cfg** > call.donotdisturb.perReg
Local Phone User Interface	
If DND is enabled, you can turn DND on or off using the Do Not Disturb key or the Do Not Disturb menu option in the Features menu	

Example Do Not Disturb Configuration

In the following example, taken from the **sip-interop.cfg** template, server-based DND has been enabled in serverFeatureControl.dnd, and `rejectBusyOnDnd` has been set to `1`—enabled—so that when you turn on DND on the phone, incoming callers will receive a busy signal.



Use the Local Contact Directory

The phones feature a contact directory you can use to store frequently used contacts.

Note that the phone follows a precedence order when looking for a contact directory. A phone will look first for a local directory in its own memory, next for a ***<MACaddress>*-directory.xml** that is uploaded to the server, and finally for a seed directory **000000000000-directory~.xml** that is included in your UC software download.

Changes you make to the contact directory from the phone are stored on the phone drive and uploaded to the provisioning server in ***<MACaddress>*-directory.xml**. This enables you to preserve a contact directory during reboots.

If you want to use the seed directory, locate **00000000000-directory~.xml** in your UC Software files on the server and remove the tilde (~) from the file name. The phone will substitute its own MAC address for <*000000000000*>.

The contact directory is the central database for several phone features including speed dial (see Configure the Speed Dial Feature), distinctive incoming call treatment (see Distinctive Incoming Call Treatment), and presence (see Use the Presence Feature. The table Use the Local Contact Directory lists the directory parameters you can configure. The CX5500 system supports up to 999 contacts. If you want to conserve phone memory, you can configure the phones to support a lower maximum number of contacts.

Tip: Deleting the Per-Phone Contact Directory

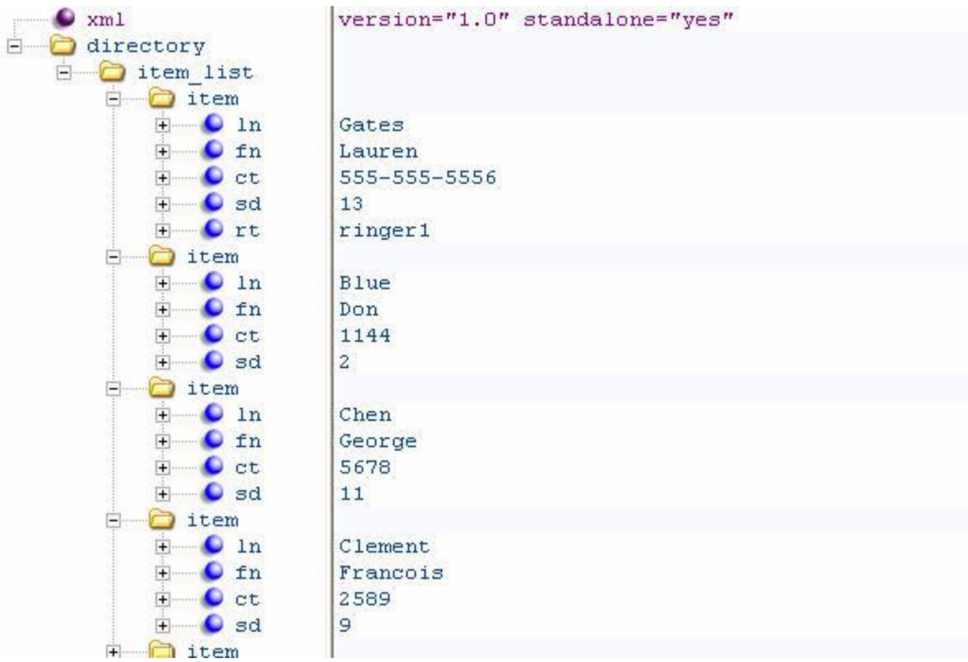
If you created a per-phone <*MACaddress*>-directory.xml for a phone and you want that phone to use a global contact directory 000000000000-directory.xml, remove the <*MACaddress*>-directory.xml file you created from the server.

Use the Local Contact Directory

Central Provisioning Server	**template** > parameter
Enable or disable the local contact directory	**features.cfg** >feature.directory.enabled
Specify if the local contact directory is read-only	**features.cfg** > dir.local.readonly
Specify the maximum number of contact entries for each phone	**features.cfg**> dir.local.contacts.maxNum
Specify whether to search the directory by first name or last name	**features.cfg** > dir.search.field
The template contact directory file	**000000000000-directory~.xml**

Example Configuration

The following illustration shows four contacts configured in a directory file.



The table Understanding the Local Contact Directory describes each of the parameter elements and permitted values that you can use in the local contact directory.

Understanding the Local Contact Directory

Element	*Definition*	*Permitted Values*
fn	**First Name**	**UTF-8 encoded string of up to 40 bytes[1]**
The contact's first name.		
ln	**Last Name**	**UTF-8 encoded string of up to 40 bytes[1]**
The contact's last name.		
ct	**Contact**	**UTF-8 encoded string containing digits (the user part of a SIP URL) or a string that constitutes a valid SIP URL**
Used by the phone to address a remote party in the same way that a string of digits or a SIP URL are dialed manually by the user. This element is also used to associate incoming callers with a particular directory entry. The maximum field length is 128 characters. Note: This field cannot be null or duplicated.		
sd	**Speed Dial Index**	**Null, 1 to 9999**
Associates a particular entry with a speed dial key for one-touch dialing or dialing from the speed dial menu. Note:		

Element	*Definition*	*Permitted Values*
lb	**Label**	**UTF-8 encoded string of up to 40 bytes**[1]
The label for the contact. Note: The label of a contact directory item is by default the label attribute of the item. If the label attribute does not exist or is Null, then the first and last names will form the label. A space is added between first and last names.		
pt	**Protocol**	**SIP, H323, or Unspecified**
The protocol to use when placing a call to this contact.		
rt	**Ring Tone**	**Null, 1 to 21**
When incoming calls match a directory entry, this field specifies the ringtone that will be used.		
dc	**Divert Contact**	**UTF-8 encoded string containing digits (the user part of a SIP URL) or a string that constitutes a valid SIP URL**
The address to forward calls to if the Auto Divert feature is enabled.		
ad	**Auto Divert**	**0 or 1**
If set to 1, callers that match the directory entry are diverted to the address specified for the divert contact element. Note: If auto-divert is enabled, it has precedence over auto-reject.		
ar	**Auto Reject**	**0 or 1**
If set to 1, callers that match the directory entry specified for the auto-reject element are rejected. Note: If auto divert is also enabled, it has precedence over auto reject.		
bw	**Buddy Watching**	**0 or 1**
If set to 1, this contact is added to the list of watched phones.		
bb	**Buddy Block**	**0 or 1**
If set to 1, this contact is blocked from watching this phone.		

[1] In some cases, this will be less than 40 characters due to UTF-8's variable bit length encoding.

Configure the Local Digit Map

The phone has a local digit map feature that, when configured, will automatically call a dialed number, eliminating the need to press the **Dial** or **Send** soft key to place outgoing calls. Note that digit maps do not apply to on-hook dialing.

Digit maps are defined by a single string or a list of strings. If a number you dial matches any string of a digit map, the call is automatically placed. If a number you dial matches no string—an impossible match—you can specify the phone's behavior. If a number ends with #, you can specify the phone's behavior, called trailing # behavior. You can also specify the digit map timeout, the period of time after you dial a number that the call will be placed. The parameter for each of these options is outlined in Configure the Local Digit Map. The configuration syntax of the digit map is based on recommendations in section 2.1.5 of RFC 3435.

Web Info: Changing the Local Digit Map on Polycom Phones

For instructions on how to modify the Local Digit Map, see *Technical Bulletin 11572: Changes to Local Digit Maps on SoundPoint IP, SoundStation IP, and Polycom VVX 1500 Phones*.

Configure the Local Digit Map

Central Provisioning Server	**template** > parameter
Apply a dial plan to dialing scenarios	**site.cfg** > dialplan.applyTo*
Specify the digit map to use for the dial plan	**site.cfg** > dialplan.digitmap
Specify the timeout for each segment of the digit map	**site.cfg** > dialplan.digitmap.timeOut
Specify the behavior if an impossible dial plan match occurs	**site.cfg** > dialplan.impossibleMatchHandling
Specify if trailing # digits should be removed from digits sent out	**site.cfg** > dialplan.removeEndOfDial
Specify the details for emergency dial plan routing	**site.cfg** > dialplan.routing.emergency.x.*
Specify the server that will be used for routing calls	**site.cfg** > dialplan.routing.server.x.*
Configure the same parameters as above for a specific registration (overrides the global parameters above)	**site.cfg** > dialplan.x.*
Specifies the time in seconds that the phone waits before dialing a number when you dial on-hook	**site.cfg** > dialplanuserDialtimeOut
Web Configuration Utility	
Specify impossible match behavior, trailing # behavior, digit map matching strings, and time-out value by navigating to **Settings > SIP** and expanding the **Local Settings** menu.	

Understand Digit Map Rules

The following is a list of digit map string rules. If you are using a list of strings, each string in the list can be specified as a set of digits or timers, or as an expression which the gateway will use to find the shortest possible match.

Digit map extension letter *R* indicates that certain matched strings are replaced. Using a *RRR* syntax, you can replace the digits between the first two *R*s with the digits between the last two *R*s. For example, **R***555***R***604***R** would replace 555 with 604. Digit map timer letter *T* indicates a timer expiry. Digit map protocol letters *S* and *H* indicate the protocol to use when placing a call. The following examples illustrate the semantics of the syntax:

- `R9R604Rxxxxxxx`—Replaces *9* with *604*
- `xxR601R600Rxx`—When applied to *1160122* gives *1160022*
- `R9RRxxxxxxx`—Remove *9* at the beginning of the dialed number (replace *9* with *nothing*)
  - For example, if a customer dials *914539400*, the first *9* is removed when the call is placed.

- `RR604Rxxxxxxx`—Prepend *604* to all seven-digit numbers (replace *nothing* with *604*)
  - For example, if a customer dials *4539400*, *604* is added to the front of the number, so a call to *6044539400* is placed.
- `xR60xR600Rxxxxxxx`—Replace any *60x* with *600* in the middle of the dialed number that matches
  - For example, if a customer dials *16092345678*, a call is placed to *16002345678*.
- `911xxx.T`—A period (.) that matches an arbitrary number, including zero, of occurrences of the preceding construct
  - For example:
    911123 with waiting time to comply with *T* is a match
    9111234 with waiting time to comply with *T* is a match
    91112345 with waiting time to comply with *T* is a match
    and the number can grow indefinitely given that pressing the next digit takes less than *T*.
- `0xxxS`—All four digit numbers starting with a *0* are placed using the SIP protocol.

Take note of the following guidelines:

- The following letters are case sensitive: *x*, *T*, *R*, *S*, and *H*.
- You must use only *, #, +, or 0–9 between the second and third *R*.
- If a digit map does not comply, it is not included in the digit plan as a valid map. That is, no match will be made.
- There is no limit to the number of *R* triplet sets in a digit map. However, a digit map that contains less than a full number of triplet sets (for example, a total of 2 *R*s or 5 *R*s) is considered an invalid digit map.
- If you use *T* in the left part of *RRR'*syntax, the digit map will not work. For example, R0TR322R will not work.

Microphone Mute

When you activate microphone mute, the Mute keys glow red . The Mute keys can be configured to mute audio and video when the CX5500 system is connected to a computer. No configuration changes can be made to the microphone mute feature when using the CX5500 system as a standalone system not connected to a computer.

Configure the Speed Dial Feature

You can link entries in your local contact directory to speed dial contacts on the phone. The speed dial feature enables you to place calls quickly using dedicated line keys or from a speed dial menu. To set up speed dial through the phone's contact directory, see Use the Local Contact Directory. Speed dial configuration is also explained briefly in Configure the Speed Dial Feature. In order to set up speed dial contacts become familiar with parameters in the table Configure the Speed Dial Feature, which identifies the directory XML file and the parameters you need to set up your speed dial contacts.

The speed dial index range is 1 to 9999.

On some call servers, enabling Presence for an active speed dial contact will display that contact’s status on the speed dial’s line key label. For information on how to enable Presence for contacts, see Use the Presence Feature.

Configure the Speed Dial Feature

Central Provisioning Server	**template** > parameter
Enter a speed dial index number in the <sd>x</sd> element in the <MAC address>-directory.xml file to display a contact directory entry as a speed dial key on the phone. Speed dial contacts are assigned to unused line keys and to entries in the phone’s speed dial list in numerical order.	
The template contact directory file	**000000000000-directory~.xml**
Local Phone User Interface	
New directory entries are assigned to the Speed Dial Index in numerical order. To assign a speed dial index to a contact, navigate go to **Contact Directory**, highlight the contact, press the **Edit** soft key, and specify a Speed Dial Index.	

Power Tip: Quick Access to the Speed Dial List

To access the Speed Dial list quickly, press the phone’s Up arrow key from the idle display.

Example Speed Dial Configuration

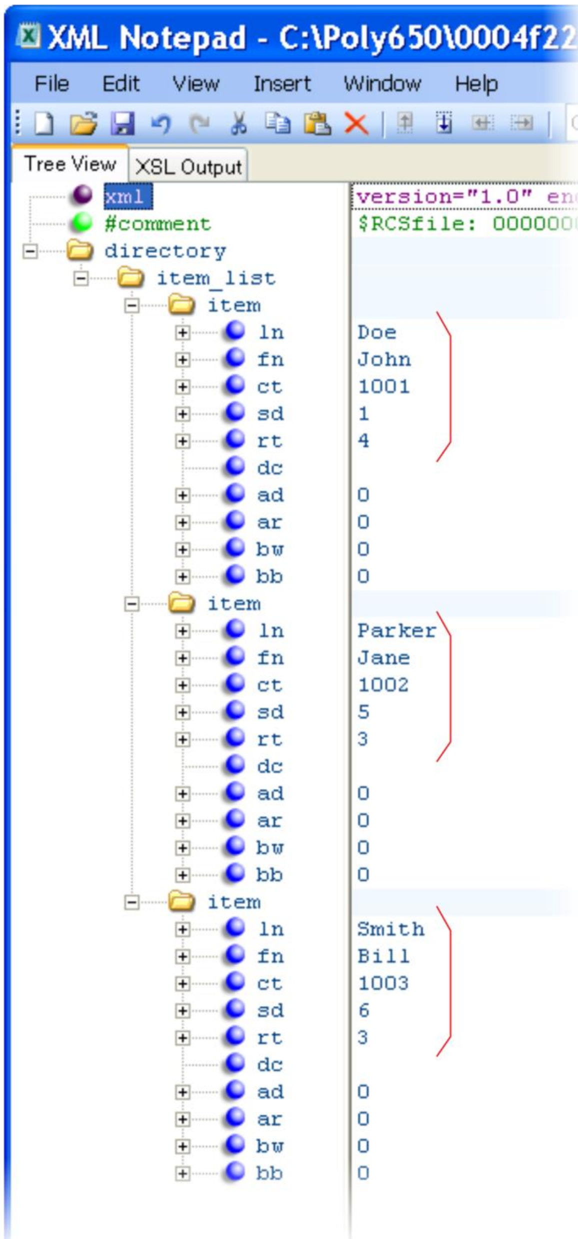
The first time you deploy and reboot the phones with UC Software, a template contact directory file named **00000000000-directory~.xml** is loaded to the provisioning server. You can edit and use this template file as a global contact directory for a group of phones or you can create your own per-phone directory file. To create a global directory, locate the **00000000000-directory~.xml** template in your UC Software files and remove the tilde (~) from the file name. When you reboot, the phone substitutes the global file with its own **<*MACaddress*>-directory.xml** which is uploaded to the server. If you want to create a per-phone directory, replace **<*000000000000*>** in the global file name with the phone’s MAC address, for example, **<*MACaddress*>-directory.xml**.

On each subsequent reboot, the phone will look for its own **<*MACaddress*>-directory.xml** and then look for the global directory. Contact directories stored locally on the phone may or may not override the **<*MACaddress*>-directory.xml** on the server depending on your server configuration. The phone will always look for a local directory or **<*MACaddress*>-directory.xml** before looking for the global directory.

For more information on how to use the template directory file **000000000000-directory~.xml**, see Use the Local Contact Directory.

Once you have renamed the directory file as a per-phone directory, enter a number in the speed dial `<sd>` field to display a contact directory entry as a speed dial contact on the phone. Speed dial entries automatically display on unused line keys on the phone and are assigned in numerical order.

The example local contact directory file shown next is saved with the phone's MAC address and shows the contact *John Doe* with extension number *1001* as speed dial entry *1* on the phone.



This configuration results in the following speed dial keys on the phone.

Set the Time and Date Display

A clock and calendar are enabled by default. You can display the time and date for your time zone in several formats, or you can turn it off altogether. You can also set the time and date format to display differently when the phone is in certain modes. For example, the display format can change when the phone goes from idle mode to an active call. You will have to synchronize the phone to the Simple Network Time Protocol (SNTP) time server. Until a successful SNTP response is received, the phone will continuously flash the time and date to indicate that they are not accurate.

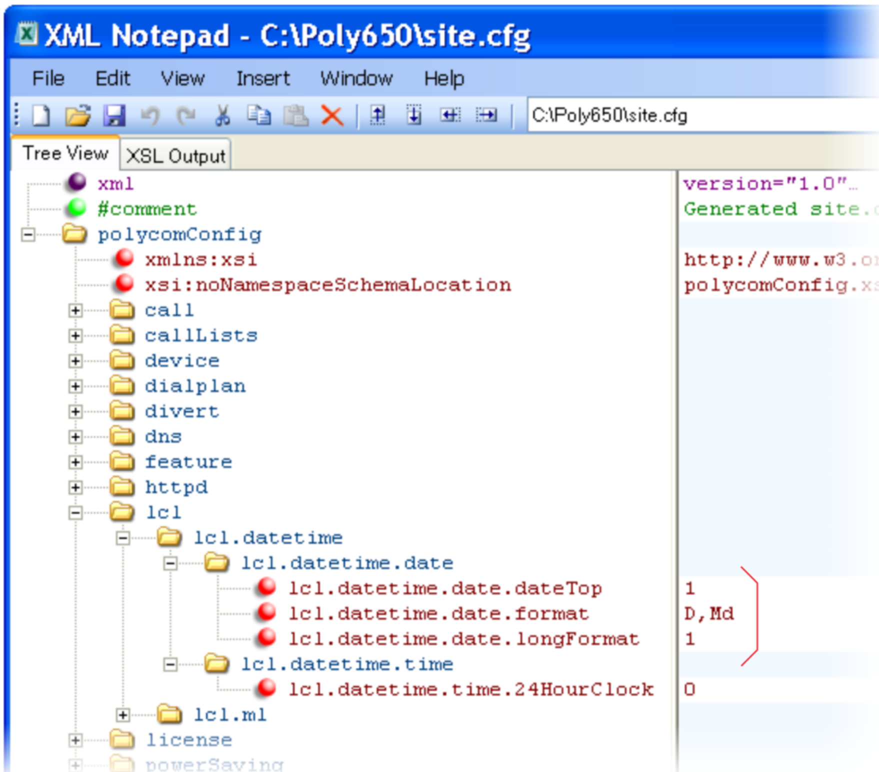
The time and date display on phones in PSTN mode will be set by an incoming call with a supported Caller ID standard, or when the phone is connected to Ethernet and you enable the turn on the date and time display.

Set the Time and Date Display

Central Provisioning Server	**template** > parameter
Turn the time and date display on or off.	**reg-advanced.cfg** and **site.cfg** > up.localClockEnabled
Set the time and date display format.	**site.cfg** > lcl.datetime.date.*
Display time in the 24-hour format	**site.cfg** > lcl.datetime.time.24HourClock
Set the basic SNTP settings and daylight savings parameters.	**site.cfg** > tcpIpApp.sntp.*
Web Configuration Utility	
To set the basic SNTP and daylight savings settings navigate to **Preferences > Date & Time**.	
Local Phone User Interface	
Basic SNTP settings can be made in the Network Configuration menu—see DHCP Menu or Network Interfaces Menu (Ethernet Menu) To set the time and date format and enable or disable the time and date display, tap **Settings > Basic > Preferences > Time & Date**.	

Example Configuration

The following illustration shows an example configuration for the time and date display format. In this illustration, the date is set to display over the time and in long format. The `D, Md` indicates the order of the date display, in this case, day of the week, month, and day. In this example, the default time format is used, or you can enable the 24-hour time display format.



Use the table Date Formats to choose values for the `lcl.datetime.date.format` and `lcl.datetime.date.longformat` parameters. The table shows values for the date Friday, August 19, 2011.

Date Formats

lcl.datetime.date.format	*lcl.datetime.date.longformat*	*Date Displayed on Phone*
dM,D	0	19 Aug, Fri
dM,D	1	19 August, Friday
Md,D	0	Aug 19, Fri
Md,D	1	August 19, Friday
D,dM	0	Fri, 19 Aug
D,dM	1	Friday, August 19

lcl.datetime.date.format	*lcl.datetime.date.longformat*	*Date Displayed on Phone*
DD/MM/YY	n/a	19/08/11
DD/MM/YYYY	n/a	19/08/2011
MM/DD/YY	n/a	08/19/11
MM/DD/YYYY	n/a	08/19/2011
YY/MM/DD	n/a	11/08/19
YYYY/MM/DD	n/a	2011/08/11

Set a Graphic Display Background

You can display a .PNG or .BMP image on the background of the touch screen. The table Set a Graphic Display Background links you to parameters and definitions in the reference section. Note that a Graphic Display Background displays across the entire screen and the time and date and line and soft key labels display over the backgrounds.

Note: Choosing a Graphic Display Background

Depending on the image you use, the graphic display background may affect the visibility of text and numbers on the phone screen. As a general rule, backgrounds should be light in shading for better phone and feature usability.

Web Info: Adding a Graphic Display Background

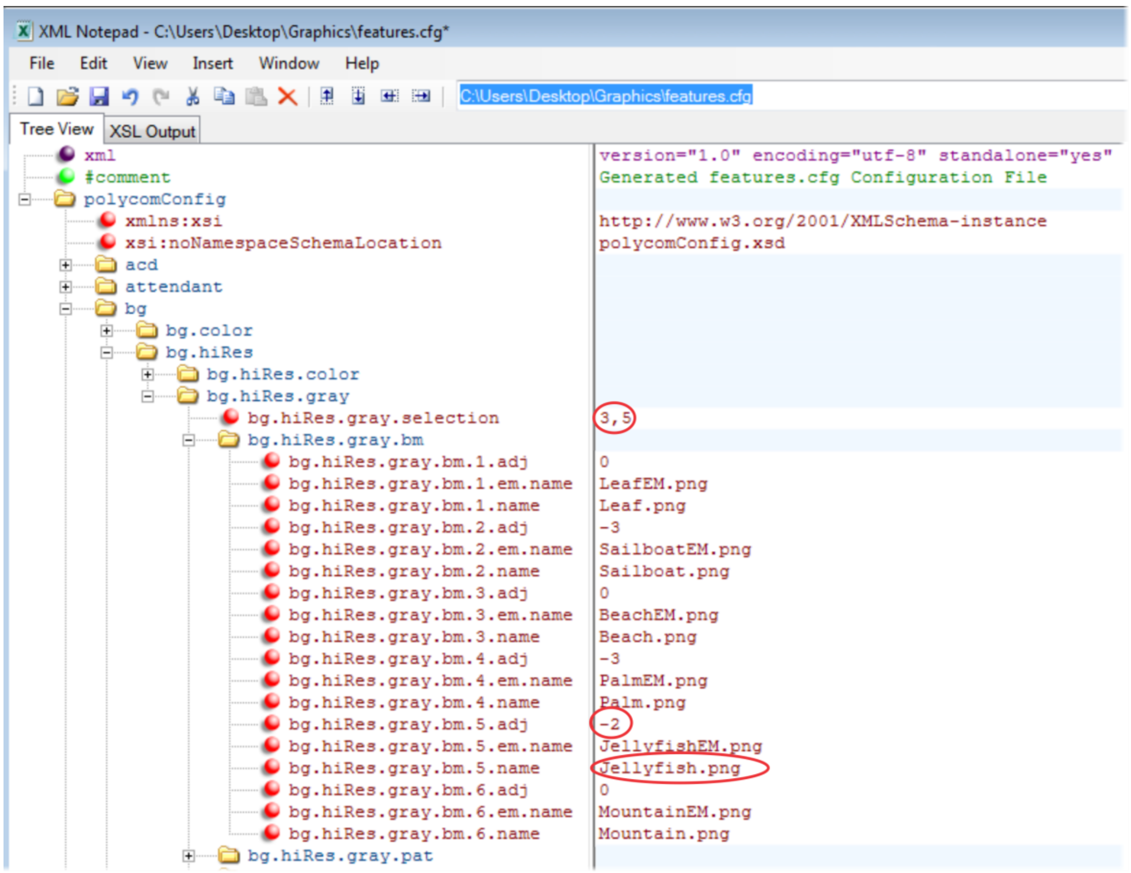
For instructions on customizing the background, see.
For detailed instructions on adding a graphic display to a phone, see *Technical Bulletin 62470: Customizing the Display Background on Your Polycom Business Media Phones*.

Set a Graphic Display Background

Central Provisioning Server	**template** > parameter
Specify a background to display for your phone type	**features.cfg** > bg.*
Modify the color of the line and soft keys	**features.cfg** > button.*
Web Configuration Utility	
Specify which background to display by navigating to **Preferences > Background**	
Local Phone User Interface	
To select a background, on the phone, navigate to **Settings > Basic > Preferences > Background > Select Background**.	

Example Graphic Display Background Configuration

This example configuration shows a background image applied to the CX5500 system. The default background in the **features.cfg** template file, specified in the `bg.hiRes.gray.selection` parameter, is set to *2,1*. Where 2 = `bg.hiRes.gray.pat.solid.*` and 1 = `bg.hiRes.gray.pat.solid.1.*` , the phone will display the solid color specified by the RBG color pattern, in this case the color named *White*. In this example, the `bg.hiRes.gray.selection` parameter has been set to *3,5*. Where 3 = `bg.hiRes.gray.bm.*` and 5 = `bg.hiRes.gray.bm.5.*` , the phone will display the image named *Jellyfish.png*. In addition, the `bg.hiRes.gray.bm.6.adj` parameter has been changed to `-2` to lighten the background image so as not to conflict with the time and date display.



This example configuration will result in the following graphic display background on the phone screen. Note that line and soft key labels will display over the background image.

Enable Automatic Off-Hook Call Placement

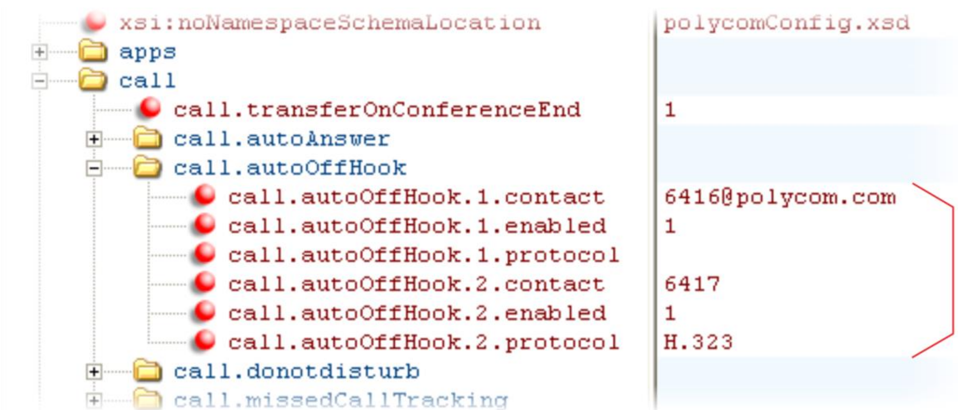
You can configure the system to automatically place a call to a specified number when you go off-hook. This feature is sometimes referred to as *hot dialing*. The phone goes off-hook when you press the New Call soft key. As shown in the table Enable Automatic Off-Hook Call Placement, you can specify an off-hook call contact and enable or disable the feature for specific line registrations.

Enable Automatic Off-Hook Call Placement

Central Provisioning Server	template > parameter
Specify the contact to dial when the phone goes off-hook	**reg-advanced** > call.autoOffHook.x.contact
Enable or disable automatic off-hook call placement on registration x	**reg-advanced** > call.autoOffHook.x.enabled

Example Automatic Off-Hook Placement Configuration

In the example configuration shown next, the automatic off-hook call placement feature has been enabled for registration 1 and registration 2. If registration 1 goes off-hook, a call is placed automatically to *6416@polycom.com*, the contact that has been specified for registration 1 in `call.autoOffHook.1.contact`. Similarly, if registration 2 goes off-hook, a call is placed automatically to `6417`..



Configure Call Hold

The purpose of call hold is to pause activity on one call so that you can use the phone for another task, for example, to place or receive another call or to search your phone's menu for information. See the table Enable Call Hold for a list of available parameters you can configure for this feature. When you place an active call on hold, a message will inform the held party that they are on hold. You can also configure a call hold alert to remind you after a period of time that a call is still on hold.

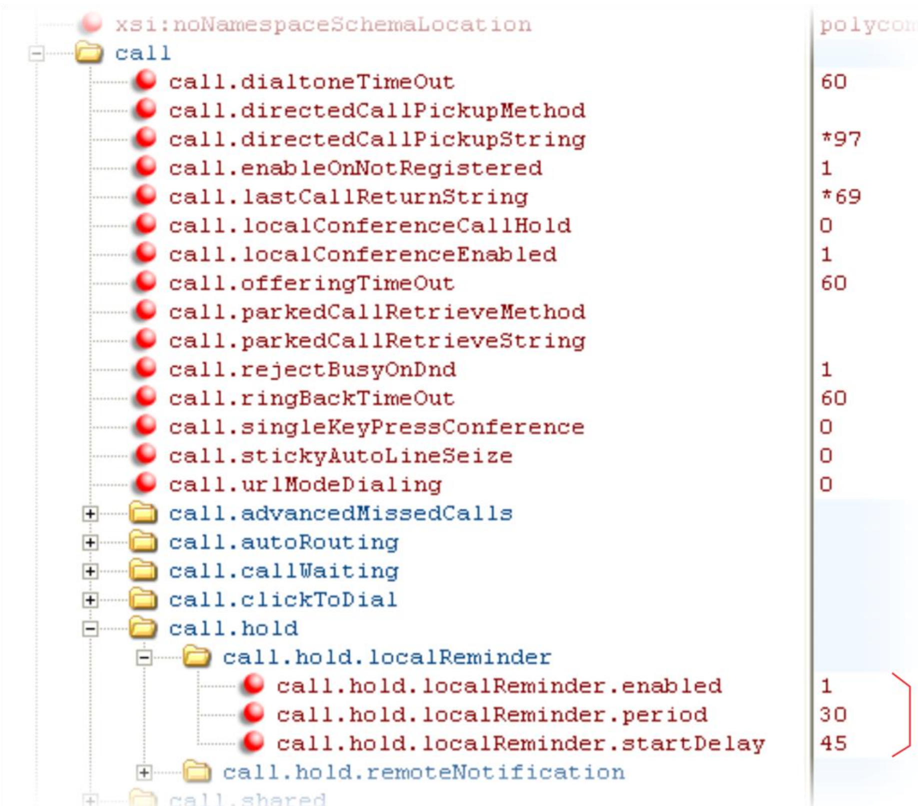
As of SIP 3.1, if supported by the call server, you can enter a music-on-hold URI. For more information, see Session Initiation Protocol Service Example - Music on Hold.

Enable Call Hold

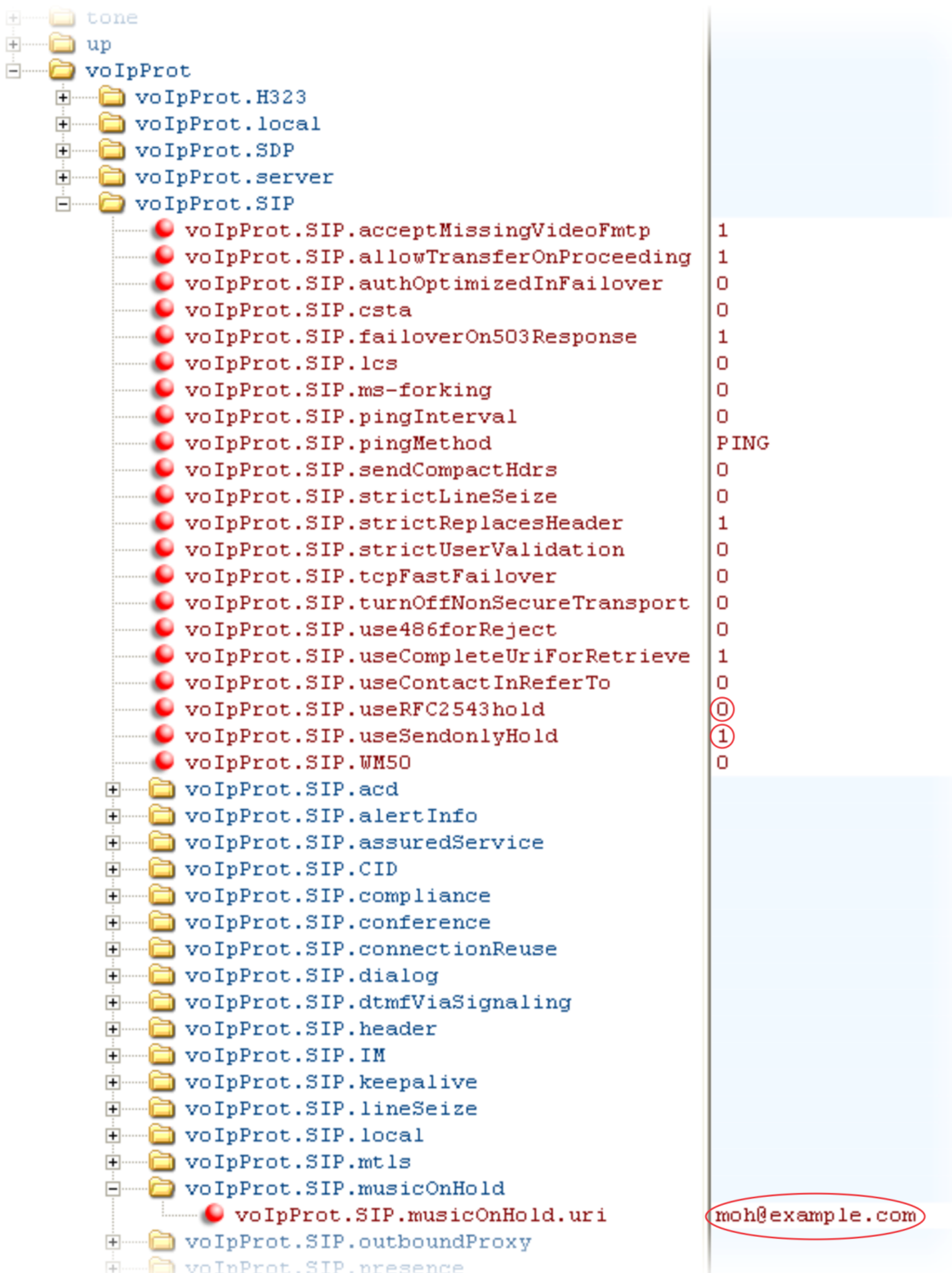
Central Provisioning Server	**template** > parameter
Specify whether to use RFC 2543 (c=0.0.0.0) or RFC 3264 (a=sendonly or a=inactive) for outgoing hold signaling	**sip-interop.cfg** > voIpProt.SIP.useRFC2543hold
Specify whether to use sendonly hold signaling	**sip-interop.cfg** > voIpProt.SIP.useSendonlyHold
Configure local call hold reminder options	**sip-interop.cfg** > call.hold.localReminder.*
Specify the music-on-hold URI	**sip-interop.cfg** > voIpProt.SIP.musicOnHold.uri
Local Phone User Interface	
Navigate to **Settings > Advanced > Administration Settings > SIP Server Configuration** to specify whether or not to use RFC 2543 (c=0.0.0.0) outgoing hold signaling. The alternative is RFC 3264 (a=sendonly or a=inactive).	

Example Call Hold Configuration

The following two illustrations show a sample configuration for the call hold feature. Both illustrations are taken from the **sip-interop.cfg** template. In the first illustration, the three `localReminder.*` parameters have been configured to play a tone to remind you of a party on hold, that the tone will begin to play 45 seconds after you put a party on hold, and that the tone will repeat every 30 seconds.



In the second illustration, the `musicOnHold.uri` parameter has been configured so the party on hold will hear music played from SIP URI *moh@example.com*.



Use Call Transfer

The Call Transfer feature enables you to transfer an existing active call to a third-party address using a Transfer soft key. For example, if party A is in an active call with party B, party A can transfer party B to party C (the third party). In this case, party B and party C will begin a new call and party A will disconnect. The table Use Call Transfer shows you how to specify call transfer behavior.

You can perform two types of call transfers:

- **Blind Transfer** Party A transfers the call without speaking to party C.
- **Consultative Transfer** Party A speaks to party C before party A transfers the call.

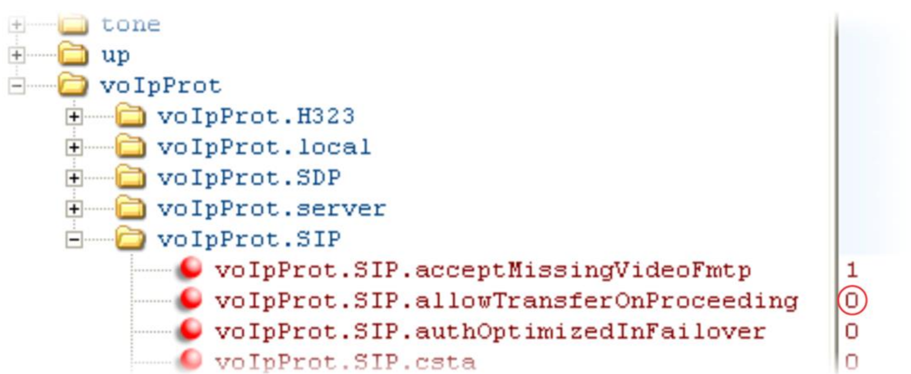
  By default, a Transfer soft key will display when party A calls Party C and Party C's phone is ringing, the proceeding state. In this case, party A has the option to complete the transfer before party C answers, which ends party A's connection to party B and C. You can disable this option so that the Transfer soft key does not display during the proceeding state. In this case, party A can either wait until party C answers or press the Cancel soft key and return to the original call.

Use Call Transfer

Central Provisioning Server	**template** > parameter
Specify whether to allow transfers while calls are in a proceeding state	**sip-interop.cfg** > voIpProt.SIP.allowTransferOnProceeding

Example Call Transfer Configuration

In the following example configuration, the parameter `allowTransferOnProceeding` has been disabled so that the Transfer soft key will not display while the third-party phone is ringing, the proceeding state. Once you have connected to the third-party, the Transfer soft key will display. If the third-party does not answer, you can press the Cancel soft key to return to the active call.



Create Local and Centralized Conferences

You can set up local or centralized conferences. Local conferences require a host phone, which processes the audio of all parties. All phones support three-party local conferencing. Alternatively, you can use an external audio bridge, available via a central server, to create a centralized conference call. Polycom recommends using centralized conferencing to host four-party conferences, though some phones do enable to host four-party conferences locally.

See the parameters in the table Create Local and Centralized Conferences to set up a conference type and the options available for each type of conference. You can specify whether, when the host of a three-party local conference leaves the conference, the other two parties remain connected or disconnected. If you want the other two parties remain connected, the phone will perform a transfer to keep the remaining parties connected. If the host of four-party local conference leaves the conference, all parties are disconnected and the conference call ends. If the host of a centralized conference leaves the conference, each remaining party remains connected. For more ways to manage conference calls, see Enable Conference Management.

Create Local and Centralized Conferences

Central Provisioning Server	**template** > parameter
Specify whether, during a conference call, the host can place all parties or only the host on hold	**sip-interop.cfg** > call.localConferenceCallHold
Specify whether or not the remaining parties can communicate after the conference host exits the conference	**sip-interop.cfg** > call.transferOnConferenceEnd
Specify whether or not all parties hear sound effects while setting up a conference	**sip-interop.cfg** > call.singleKeyPressConference
Specify which type of conference to establish and the address of the centralized conference resource	**sip-interop.cfg** > voIpProt.SIP.conference.address

Enable Conference Management

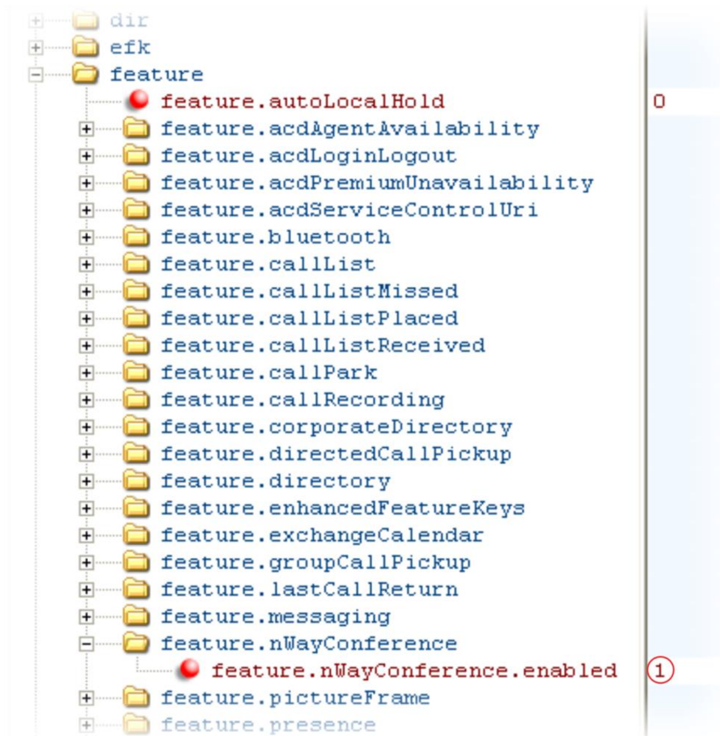
This feature enables you to add, hold, mute, and remove conference participants, as well as obtain additional information about participants. Use the parameters listed in the table Manage Conferences to configure how you want to manage conferences.

Manage Conferences

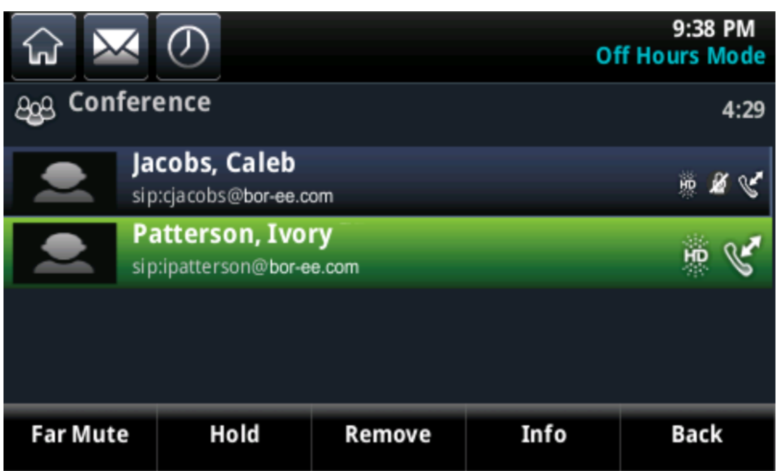
Central Provisioning Server	**template** > parameter
Enable or disable the conference management feature.	**features.cfg** > feature.nWayConference.enabled

Example Conference Management Configuration

The following example shows you how to enable the conference management feature in the **features.cfg** file.



When you enable conference management, a **Manage** soft key will display on the phone during a conference. When you press the **Manage** soft key, the **Manage Conference** screen, shown next, will display with soft keys you can use to manage conference participants.



Configure Call Forwarding

The phone provides a flexible call forwarding feature that enables you to forward incoming calls to another destination. You can apply call forwarding in the following ways:

- To all calls
- To incoming calls from a specific caller or extension
- When your phone is busy
- When Do Not Disturb is enabled
- When the phone has been ringing for a specific period of time
- You can have incoming calls forwarded automatically to a predefined destination you choose or you can manually forward calls to a destination.

You will find parameters for all of these options in the table Configure Call Forwarding.

To enable server-based call forwarding, you must enable the feature on both a registered phone and on the server and the phone is registered. If you enable server-based call forwarding on one registration, other registrations will not be affected. Server-based call forwarding disables local Call Forward and DND features.

Troubleshooting: Call Forwarding Does Not Work on My Phone

The server-based and local call forwarding features do not work with the Shared Call Appearance (SCA) and Bridged Line Appearance (BLA) features. If you have SCA or BLA enabled on your phone, disable the feature before you can use call forwarding.

The call server uses the Diversion field with a SIP header to inform the phone of a call’s history. For example, when you enable call forwarding, the Diversion header allows the receiving phone to indicate who the call was from, and the phone number it was forwarded from.

Configure Call Forwarding

Central Provisioning Server	**template** > parameter
Enable or disable server-based call forwarding	**sip-interop.cfg** > voIpProt.SIP.serverFeatureControl.cf
Enable or disable local call forwarding behavior when server-based call forwarding is enabled	**sip-interop.cfg** > voIpProt.SIP.serverFeatureControl.localProcessing.cf
Enable or disable the display of the Diversion header and the order in which to display the caller ID and number	**sip-interop.cfg** > voIpProt.SIP.header.diversion.*
Set all call diversion settings including a global forward-to contact and individual settings for call forward all, call forward busy, call forward no-answer, and call forward do-not-disturb	**site.cfg** > divert.*
Enable or disable server-based call forwarding as a per-registration feature	**reg-advanced.cfg** > reg.x.fwd.*

Web Configuration Utility
To set all call diversion settings navigate to **Settings > Lines**, select a line from the left pane, and expand the **Call Diversion** menu.
Local Phone User Interface
To enable and set call forwarding from the phone, navigate to **Settings > Features > Forward**.

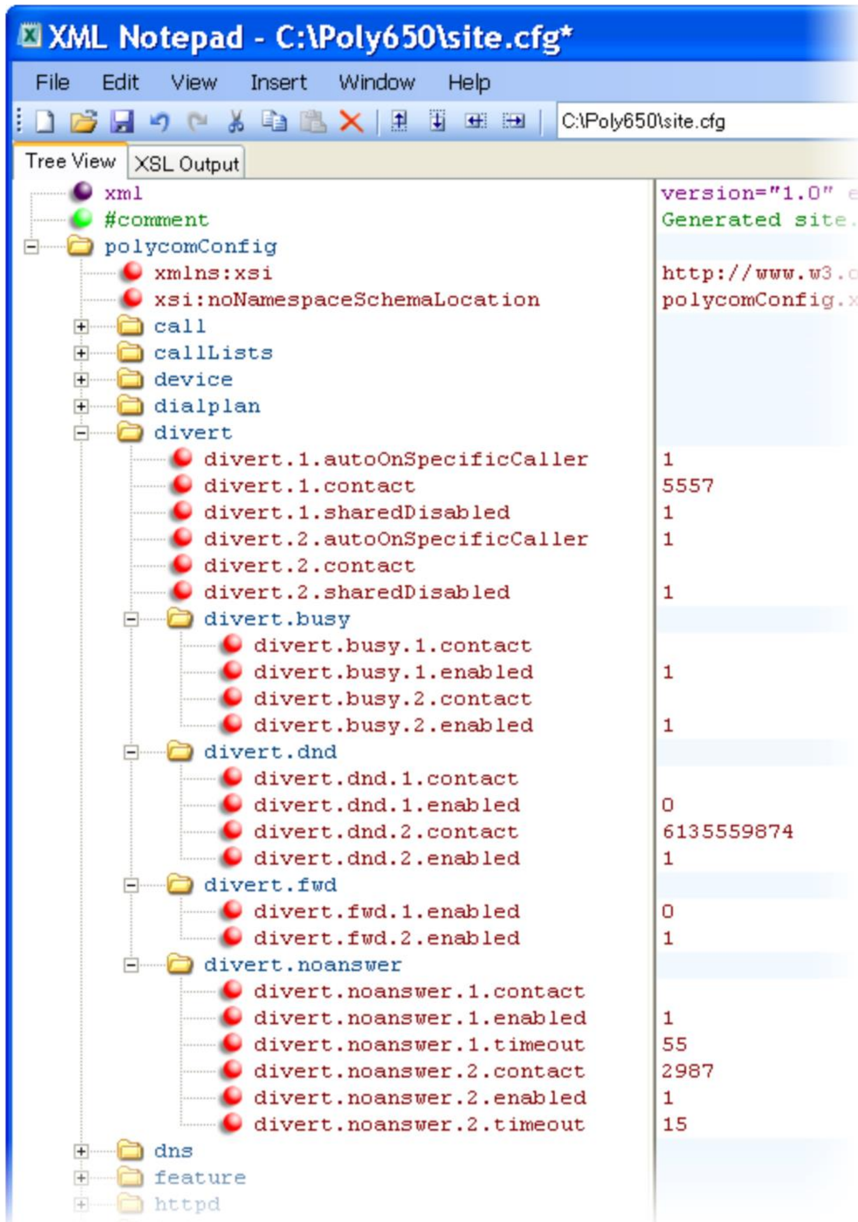
Example Call Forwarding Configuration

In the example configuration shown next, the call forwarding parameters for registration 1 have been changed from the default values. The forward-always contact for registration 1 is `5557` and this number will be used if the parameters `divert.busy`, `divert.dnd`, or `divert.noanswer` are not set. Parameters you set in those fields will override `divert.1.contact`.

To enable these three divert options for each registration, enable the `divert.fwd.x.enabled` parameter and the `.enabled` parameter for each of the three forwarding options you want to enable.

In this example, `divert.fwd.1.enabled` has been disabled; all calls to registration 1 will be diverted to `5557` and you do not have the option of enabling any of the three forwarding options on the phone. The three divert options are enabled for registration 2 in the `divert.fwd.2.enabled` parameter, giving you the option to enable or disable any one of the three forwarding options on the phone.

When do not disturb (DND) is turned on, you can set calls to registration 2 to be diverted to `6135559874` instead of `5557`. The parameter `divert.noanswer.2.enabled` is enabled so that, on the phone, you can set calls to registration 2 that ring for more than 15 seconds, specified in `divert.noanswer.2.timeout`, to be diverted to `2987`, as set in `divert.noanswer.2.contact`.



Configure Lync Call Forwarding

The following types of call forwarding are available on Lync-enabled Polycom phones:

- Disable Call Forwarding
- Forward to a contact

- Forward to voicemail

No parameters are needed to enable call forwarding on Lync-enabled phones.

Configure Directed Call Pick-Up

This feature enables you to pick up incoming calls to another phone by dialing the extension of that phone. This feature requires support from a SIP server and setup of this feature depends on the SIP server. For example, while some SIP servers implement directed call pick-up using a star-code sequence, others implement the feature using network signaling. The table Configure Directed Call Pickup lists the configuration parameters for the directed call pick-up feature.

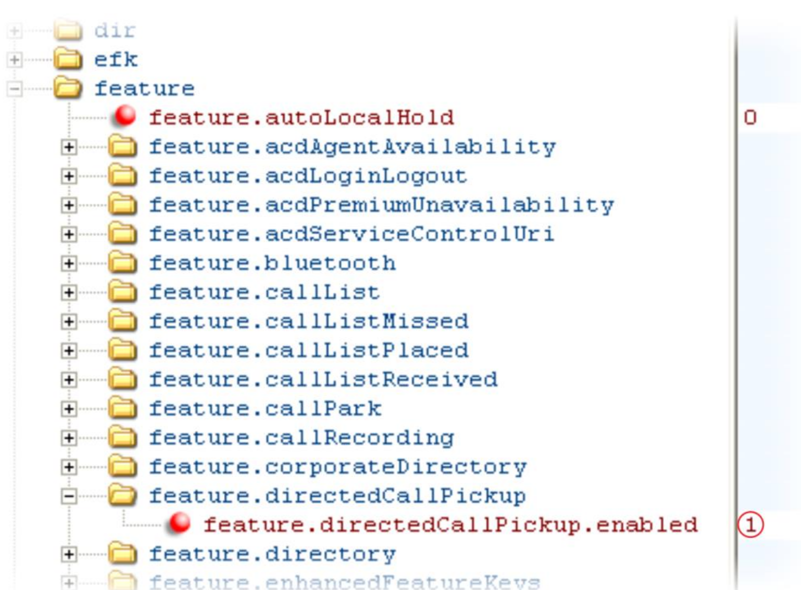
Configure Directed Call Pickup

Central Provisioning Server	template > parameter
Turn this feature on or off	**features.cfg** > feature.directedCallPickup.enabled
Specify the star code to initiate a directed call pickup	**sip-interop.cfg** > call.directedCallPickupString

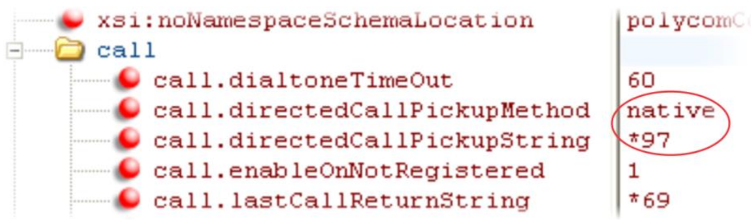
Example Directed Call Pickup Configuration

The configuration parameters for the directed call pickup feature are located in two template files. You enable directed call pickup in the **features.cfg** template file and configure the feature using the **sip-interop.cfg** file.

In the following configuration example, the directed call pickup feature has been enabled in the **features.cfg** template file:



Once directed call pickup is enabled, you can configure the feature using parameters located in the **sip-interop.cfg** template file. In the following illustration, the pickup method has been set to `native`, which means that the server is used for directed call pickup instead of the `PickupString`. If the pickup method was set to `legacy`, the pickup string `*97` would be used by default. The pickup string can be different for different call servers, check with your call server provider if you configure legacy mode directed call pickup.



When you enable directed call pickup, the phone displays a **Pickup** soft key when you go off-hook. When you press the **Pickup** soft key, the **Directd** soft key will display.

Enable Group Call Pickup

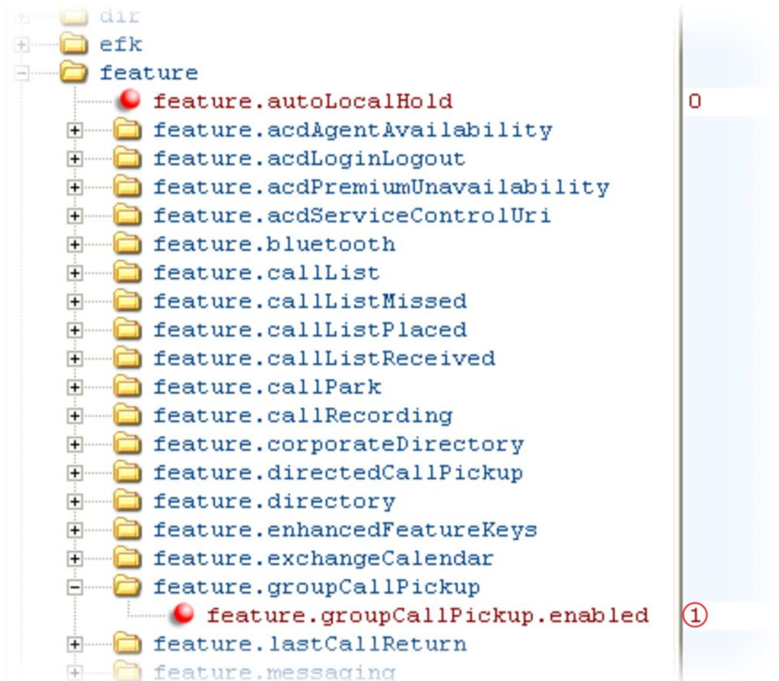
This feature enables you to pick up incoming calls to any phone within a predefined group of phones, without dialing the extension of another phone. The parameter to enable this feature is shown in the table Enable Group Call Pickup. This feature requires support from a SIP server and setup of this feature depends on the SIP server. For example, while some SIP servers implement group call pick-up using a particular star-code sequence, others implement the feature using network signaling.

Enable Group Call Pickup

Central Provisioning Server	template > parameter
Turn this feature on or off	**features.cfg** > feature.groupCallPickup.enabled

Example Group Call Pickup Configuration

The following illustration shows you how to enable the group call pickup feature in the **features.cfg** template.



When you enable the group call pickup, the phone will display a **Pickup** soft key when you go off-hook. If you select **Pickup**, the **Group** soft key is displayed.

After you press the **Group** soft key, the phone performs a just-in-time subscription request to the fixed address `<groupcallpickup@<*yourCallServerDomain*>` for dialog details with which it can pick up the original caller using a replaces header in a new INVITE.

Configure Call Park and Retrieve

This feature is available as Open SIP. If you want to use the Call Park feature available with Lync Server, see Feature Profile 84538. You can park an active call and retrieve parked calls from any phone. Whereas call hold keeps the held call on the same line, call park moves the call to a separate address where the call can be retrieved by any phone. This feature requires support from a SIP server and setup of this feature depends on the SIP server. For example, while some SIP servers implement group call pick-up using a particular star-code sequence, others implement the feature using network signaling. See the table Configure Call Park and Retrieve for parameters you can configure.

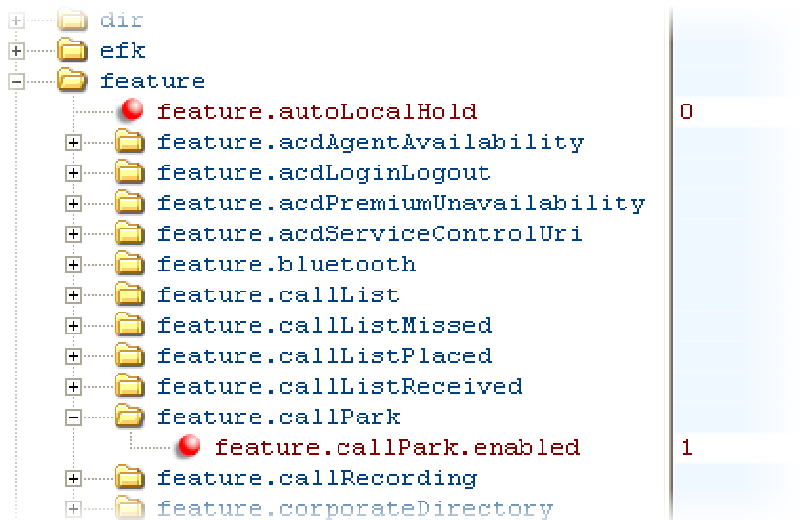
Configure Call Park and Retrieve for Open SIP

Central Provisioning Server	template > parameter
Enable or disable call park and retrieve	**features.cfg** > feature.callPark.enabled
Specify the star code used to retrieve a parked call	**sip-interop.cfg** > call.parkedCallRetrieveString

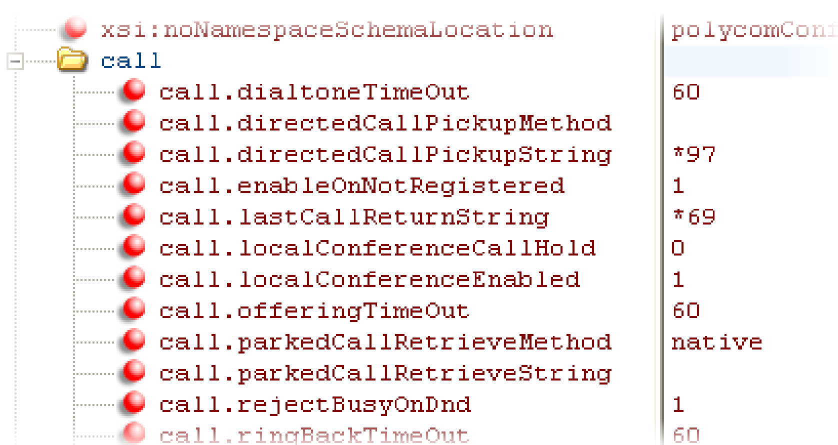
Example Call Park and Retrieve Configuration

The configuration parameters for the call park and retrieve feature are located in two template files. You can enable the feature using the **features.cfg** template file and configure the feature using the **sip-interop.cfg** file.

In the following configuration example, the call park feature has been enabled in the **features.cfg** template file.



You can configure the call park and call retrieve feature using parameters located in the **sip-interop.cfg** template file. The following illustration shows that the parked call retrieve method has been set to `native`, meaning that the phone will use SIP INVITE with the Replaces header. The method can also be set to `legacy`, meaning that the phone will use the `call.parkedCallRetrieveString` star code to retrieve the parked call.



When the call park and retrieve feature is enabled, the Park soft key displays when you are in a connected call. To park the call, press the **Park** soft key.

To retrieve a parked call, go off-hook and press the **Retrieve** soft key, or tap **New Call** soft key, enter the call orbit number, and tap **Call**.

Enable Last Call Return

The phone supports redialing of the last received call. The table Enable Last Call Return shows you the parameters to enable this feature. This feature requires support from a SIP server. With many SIP servers, this feature is implemented using a particular star code sequence. With some SIP servers, specific network signaling is used to implement this feature.

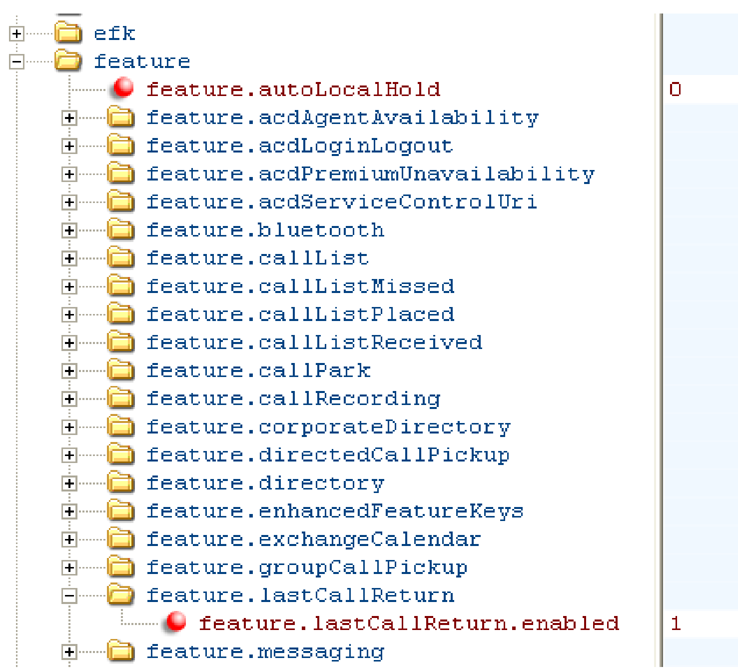
Enable Last Call Return

Central Provisioning Server	template > parameter
Enable or disable last call return	**features.cfg** > feature.lastCallReturn.enabled
Specify the string sent to the server for last-call-return	**sip-interop.cfg** > call.lastCallReturnString

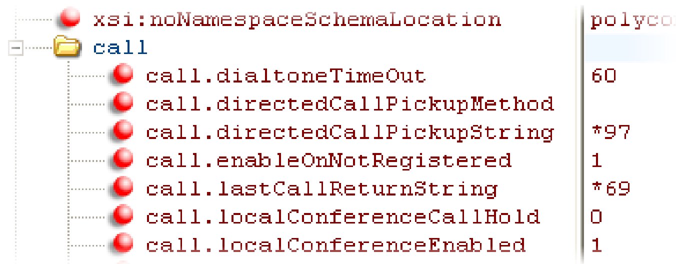
Example Configuration for Last Call Return

The configuration parameters for last call return feature are located in two template files. You can enable the feature using the **features.cfg** template file and configure the feature using the **sip-interop.cfg** file.

In the following configuration example, the last call return feature has been enabled in the **features.cfg** template file:



Once last call return is enabled, you can configure the feature using parameters located in the **sip-interop.cfg** template file. The following shows the default value for the `call.lastCallReturnString` parameter. The last call return string value depends on the call server you use. Consult with your call server provider for the last call return string.



When you enable the last call return feature, the phone displays an **LCR** soft key when it goes off-hook, as shown next. When you press the **LCR** soft key, you place a call to the phone address that last called you.

When you select **Last Call Return**, you place a call to the phone address that last called you.

Set Up Advanced Phone Features

After you set up your Polycom phones with a default configuration on the network, phone users will be able to place and receive calls; however, you may want to make some changes to optimize your configuration for your organization and user's needs. Polycom provides basic and advanced features that you can configure for the phones. This section will show you how to configure all available advanced phone features, call server features, and Polycom and third-party applications.

Before you begin configuring phone features, take the time to read the short introductory section Read the Feature Parameter Tables. This section provides important information you need to know in order to successfully perform configuration changes.

This section shows you how to make configuration changes for the following advanced features:

- Assign Multiple Line Keys Per Registration   Assign multiple line keys to a single registration.
- Enable Multiple Call Appearances   All phones support multiple concurrent calls. You can place any active call on hold to switch to another call.
- Set the Phone Language   All phones have multilingual user interfaces.
- Synthesized Call Progress Tones   Match the phone's call progress tones to a region.
- Configure Real-Time Transport Protocol Ports   Phone treat all real time transport protocol (RTP) streams as bi-directional from a control perspective, and expect that both RTP endpoints will negotiate the respective destination IP addresses and ports.
- Configure Network Address Translation   Phones can work with certain types of network address translation (NAT).
- Use the Corporate Directory   You can configure the phone to access your corporate directory if it has a standard LDAP interface. This feature is part of the Productivity Suite. Active Directory, OpenLDAP, Microsoft ADAM, and SunLDAP are currently supported.
- Configure Enhanced Feature Keys   Enables you to redefine soft keys to suit your needs. In SIP 3.0, this feature required a license key. In later releases, no license key is required.
- Configure Soft Keys   Enables you to create your own soft keys, and display them with or without the standard soft keys.
- Enable the Power Saving Feature   Enable and set hours for the power-saving feature.
- Configure Group Paging   Send one-way page broadcasts.
- Enable Bridged Line Appearance   Allows a line extension or phone number to appear on multiple users' phones. This feature requires call server support.
- Enable Voicemail Integration   Enables access to compatible voice mail servers.
- Enable Multiple Registrations   The CX5500 system supports multiple registrations.
- Set Up Server Redundancy   Phones support server redundancy to ensure the continuity of phone service when the call server is offline for maintenance, fails, or the connection between the phone and server fails.
- DNS SIP Server Name Resolution   Enter the DNS name for a proxy/registrar address.
- Use the Presence Feature   Enables you to monitor the status of other users/devices, and for other users/devices to monitor you. This feature requires call server support.

- Configure the Static DNS Cache   Set up a cache for DNS information and provide for negative caching.
- Display SIP Header Warnings   Displays a pop-up warning message to the users from a SIP header message.
- Quick Setup of the CX5500 System   Provides a simplified interface to enter provisioning server parameters while your phone boots.
- Provisional Polling of the CX5500 System   You can set the phones to automatically check for software downloads using a random schedule or through a predefined schedule.

This section also shows you how to make configuration changes to support the following Polycom and third-party applications:

- Set Up Microsoft Lync Server 2010 and 2013   You can use the CX5500 with Microsoft Lync Server 2010 to immediately share ideas and information with business contacts. This feature requires call server support.
- Enable Microsoft Exchange Calendar Integration   Enables users to manage meetings and reminders with your CX5500 system, and enables you to dial in to conference calls. This feature requires Microsoft Exchange Calendar Integration.

Assign Multiple Line Keys Per Registration

You can assign a single registered phone line address to multiple line keys on the CX5500 system. See the table Multiple Line Keys Per Registration for the parameter you need to set. This feature can be useful for managing a high volume of calls to a line. This feature is one of several features associated with *Flexible Call Appearances*. See the following table for the maximum number of line keys per registration for each phone model, and for definitions of all features associated with Flexible Call Appearances.

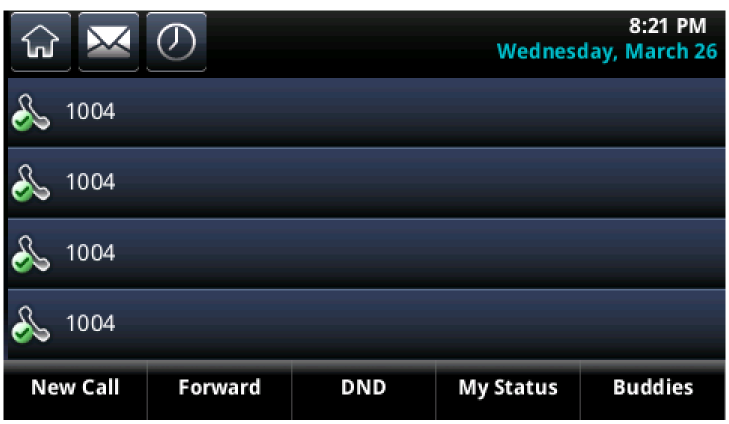
Multiple Line Keys Per Registration

Central Provisioning Server	**template** > parameter
Specify the number of line keys to use for a single registration	**reg-advanced.cfg** > reg.x.lineKeys
Web Configuration Utility	
To assign the number of line keys per registration, navigate **Settings > Lines**, select the number of lines from the left pane, expand **Identification**, and edit Number of **Line Keys**	
Local Phone User Interface	
Assign the number of line keys per registration by navigating to **Settings > Advanced > Admin Settings > Line Configuration > Line x > Line Keys > Num Line Keys**.	

Example Configuration

The following illustration shows you how to enable four line keys with the same registered line address. In this example, four line keys are configured with registration address *1004*.

The phone displays the registered line address *1004* on four line keys, as shown next.



Enable Multiple Call Appearances

You can enable each registered CX5500 system line to support multiple concurrent calls and have each concurrent call display on the phone's user interface. For example, you can place one call on hold, switch to another call on the same registered line, and have both calls display. As shown in the table Enable Multiple Call Appearances, you can set the maximum number of concurrent calls per registered line and the default number of calls per line key.

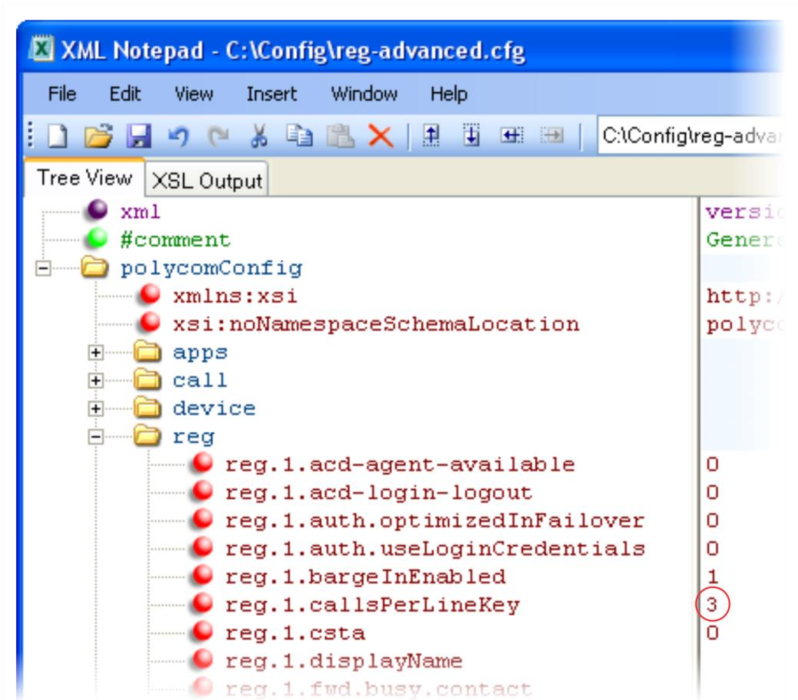
This feature is one of several features associated with *Flexible Call Appearances*. If you want to enable multiple line keys per registration, see Assign Multiple Line Keys Per Registration. Note that if you assign a registered line to multiple line keys, the default number of concurrent calls will apply to all line keys. See the following table if you want use multiple registrations on a phone, and for definitions of all features associated with Flexible Call Appearances. Use this table to customize the number of registrations, line keys per registration, and concurrent calls.

Enable Multiple Call Appearances

Central Provisioning Server	**template** > parameter
Set the default number of concurrent calls for all line keys	**reg-basic.cfg** > call.callsPerLineKey
Override the default number of calls per line key for a specific line	**reg-advanced.cfg** > reg.x.callsPerLineKey
Web Configuration Utility	
To set the default number of concurrent calls a line key, navigate to **Settings > SIP**, expand **Local Settings**, and edit **Calls Per Line Key**. To override the number of concurrent calls for a specific line, navigate to **Settings > Lines**, select the line to modify from the left pane, expand **Identification**, and edit **Calls Per Line**.	
Local Phone User Interface	
Assign the default number of concurrent calls per line by navigating to **Settings > Advanced > Admin Settings > Line Configuration > Calls Per Line Key (navigate to Line Configuration > Line *X* > Line Keys > Calls Per Line Key** to change the calls per line for only line *x*).	

Example Multiple Call Appearances Configuration

The following illustration shows that in the **reg-advanced.cfg** template you can enable line 1 on your phone with three call appearances.



Once you have set the `reg.1.callsPerLineKey` parameter to `3`, you can have three call appearances on line 1. By default, additional incoming calls will be automatically forwarded to your voicemail. If you have more than two call appearances, a call appearance counter displays at the top-right corner of your phone's screen.

The following table describes the features associated with Flexible Call Appearances. Use the table to understand how you can organize registrations, line keys per registration, and concurrent calls per line key.

Flexible Call Appearances Features

Feature	*Description*	*Limit*
Registrations	Maximum number of user registrations	16
Line Keys	Maximum number of line keys	16
Line Keys per Registration	Maximum number of line keys per user registration	16
Calls per Line Key	Maximum number of concurrent calls per line key	24
Concurrent Calls, including Conference Legs *****	Runtime maximum number of concurrent calls (Number of conference participants minus the moderator)	24 (2)

* Note that each conference leg counts as one call. The total number of concurrent calls in a conference indicated in this table includes all conference participants *minus* the moderator.

Set the Phone Language

You can select the language that displays on the phone using the parameters in the table Set the Phone Language. Each language is stored as a language file in the **SoundPointIPLocalization** folder. This folder is included with the Polycom UC Software you downloaded to your provisioning server. If you want to edit the language files, use a Unicode-compatible XML editor such as XML Notepad 2007 and familiarize yourself with the guidelines on basic and extended character support, see <ml/>.

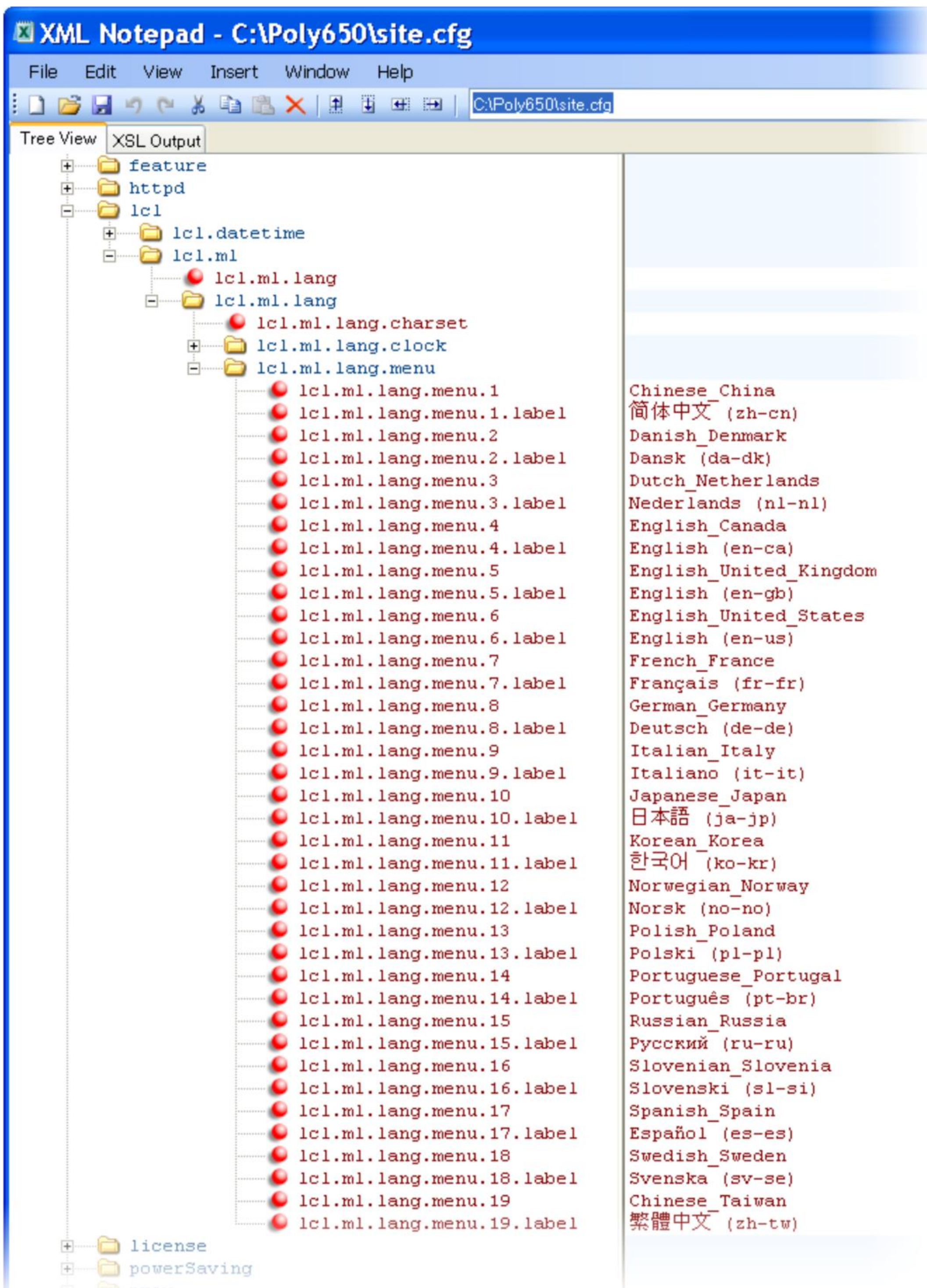
The Polycom phones support major western European languages. The CX5500 system supports the following languages: Simplified Chinese, Traditional Chinese, Danish, Dutch, English, French, German, Italian, Japanese, Korean, Norwegian, Polish, Brazilian Portuguese, Russian, Slovenian, International Spanish, and Swedish.

Set the Phone Language

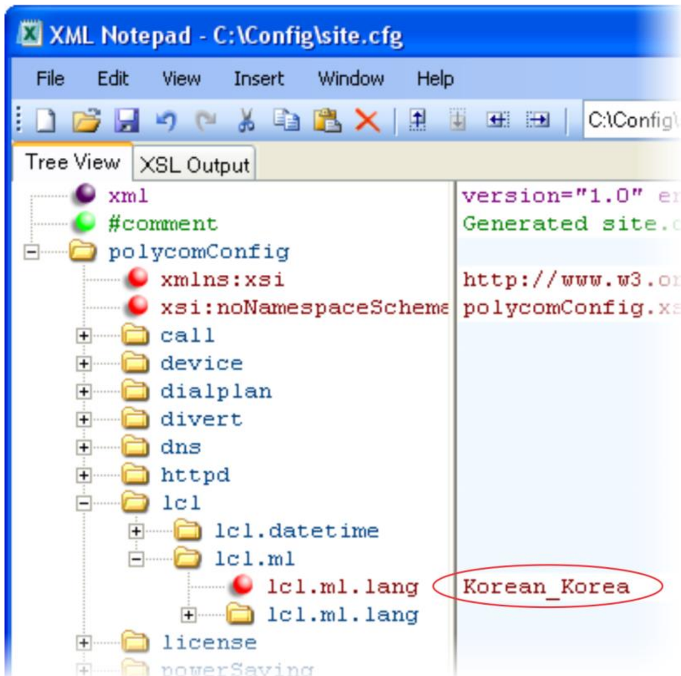
Central Provisioning Server	**template** > parameter
Obtain the parameter value for the language you want to display on the phone	**site.cfg** > lcl.ml.lang.menu.*
Specify the language used on the phone's display screen	**site.cfg** > lcl.ml.lang
Web Configuration Utility	
To change the language of the phone's display screen, navigate to **Prefences > Additional Preferences**, change **Phone Language**, and click **Add > Save**.	
Local Phone User Interface	
To change the language of the phone's display screen, navigate to **Settings > Basic > Preferences > Language**.	

Example Phone Language Configuration

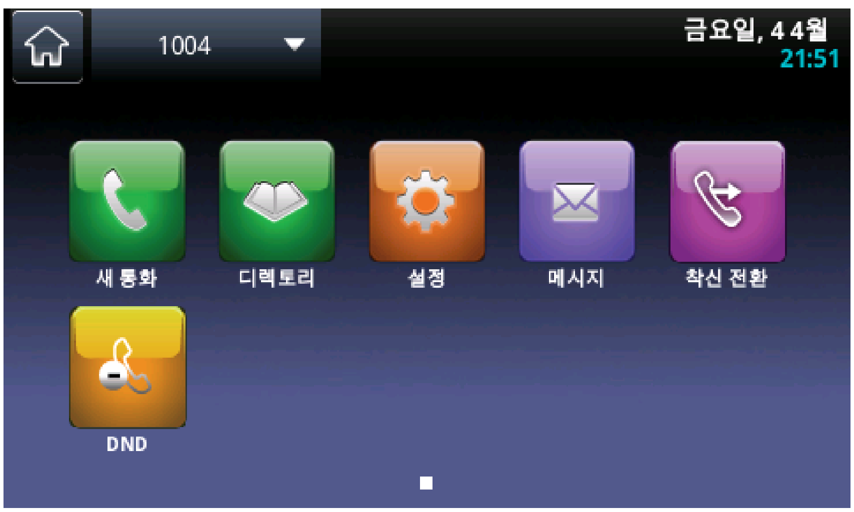
The following illustration shows you how to change the phone language. Locate the language you want the phone to display in the site.cfg template in `lcl.ml.lang.*` menu.



From the list, select the language you want to use and enter it in `lcl.ml.lang`. In the following example, the phone is set to use the Korean language.



Once configured, the phone uses Korean characters, as shown next.



Synthesized Call Progress Tones

The CX5500 system plays call signals and alerts, called call progress tones, such as busy signals, ringback sounds, and call waiting tones. The built-in call progress tones on your phone match standard North American tones. If you would like to customize the phone's call progress tones to match the standard tones in your region, contact Polycom Support.

Configure Real-Time Transport Protocol Ports

You can configure the phone to filter incoming RTP packets. You can filter the packets by IP address, or by port. For greater security, you can also configure RTP settings to reject packets arriving from a non-negotiated IP address or from an unauthorized source. You can reject packets that the phone receives from a non-negotiated IP address or a non-negotiated port.

You can configure the phone to enforce symmetric port operation for RTP packets. When the source port is not set to the negotiated remote sink port, arriving packets can be rejected.

You can also fix the phone's destination transport port to a specified value regardless of the negotiated port. This can be useful for communicating through firewalls. When you use a fixed transport port, all RTP traffic is sent to and arrives on that specified port. Incoming packets are sorted by the source IP address and port, which allows multiple RTP streams to be multiplexed.

You can specify the phone's RTP port range. Since the phone supports conferencing and multiple RTP streams, the phone can use several ports concurrently. Consistent with RFC 1889, the next-highest odd-numbered port is used to send and receive RTP. The table Configure Real-Time Transport Protocol provides a link to the reference section.

The phone is compatible with RFC 1889 - RTP: A Transport Protocol for Real-Time Applications - and the updated RFCs 3550 and 3551. Consistent with RFC 1889, the phone treats all RTP streams as bi-directional from a control perspective and expects that both RTP endpoints will negotiate the respective destination IP addresses and ports. This allows real-time transport control protocol (RTCP) to operate correctly even with RTP media flowing in only a single direction, or not at all.

Configure Real-Time Transport Protocol Ports

Central Provisioning Server	**template** > parameter
Filter RTP packets by port	**site.cfg** > tcpIpApp.port.rtp.filterByPort
Force-send packets on a specified port	**site.cfg** > tcpIpApp.port.rtp.forceSend
Set the starting port for RTP packet port range	**site.cfg** > tcpIpApp.port.rtp.mediaPortRangeStart
Web Configuration Utility	
Filter RTP packets by IP address, by port, force-send packets on a specified port, and set the port range start by navigating to **Settings > Network > RTP**.	

Example Real-Time Transport Protocol Configuration

The following illustration shows the default real-time transport protocol settings in the **site.cfg** template file. The parameter `tcpIpApp.port.rtp.filterByIp` is set to 1 so that the phone will reject RTP packets sent from non-negotiated IP addresses. The parameter `tcpIpApp.port.rtp.filterByPort` is set to 0 so that RTP packets sent from non-negotiated ports will not be rejected. Enter a value in the `tcpIpApp.port.rtp.forceSend` parameter to specify the port that all RTP packets will be sent to and received from. The parameter `tcpIpApp.port.rtp.mediaPortrangeStart shows the default` starting port *2222* for RTP packets. The starting port must be entered as an even integer.

Configure Network Address Translation

The phone can work with certain types of Network Address Translation (NAT). NAT enables a local area network (LAN) to use one set of IP addresses for internal traffic and another set for external traffic. The phone’s signaling and Real-Time Transport Protocol (RTP) traffic use symmetric ports. You can configure the external IP address and ports used by the NAT on the phone’s behalf on a per-phone basis. The table Network Access Translation lists each of the parameters you can configure. Note that the source port in transmitted packets is the same as the associated listening port used to receive packets.

Network Access Translation

Central Provisioning Server	**template** > parameter
Specify the external NAT IP address	**sip-interop.cfg** > nat.ip
Specify the external NAT keepalive interval	**sip-interop.cfg** > nat.keepalive.interval
Specify the external NAT media port start	**sip-interop.cfg** > nat.mediaPortStart
Specify the external NAT signaling port	**sip-interop.cfg** > nat.signalPort
Web Configuration Utility	
Specify the external NAT IP address, the signaling port, the media port start, and the keepalive interval by navigating to **Settings > Network > NAT**.	

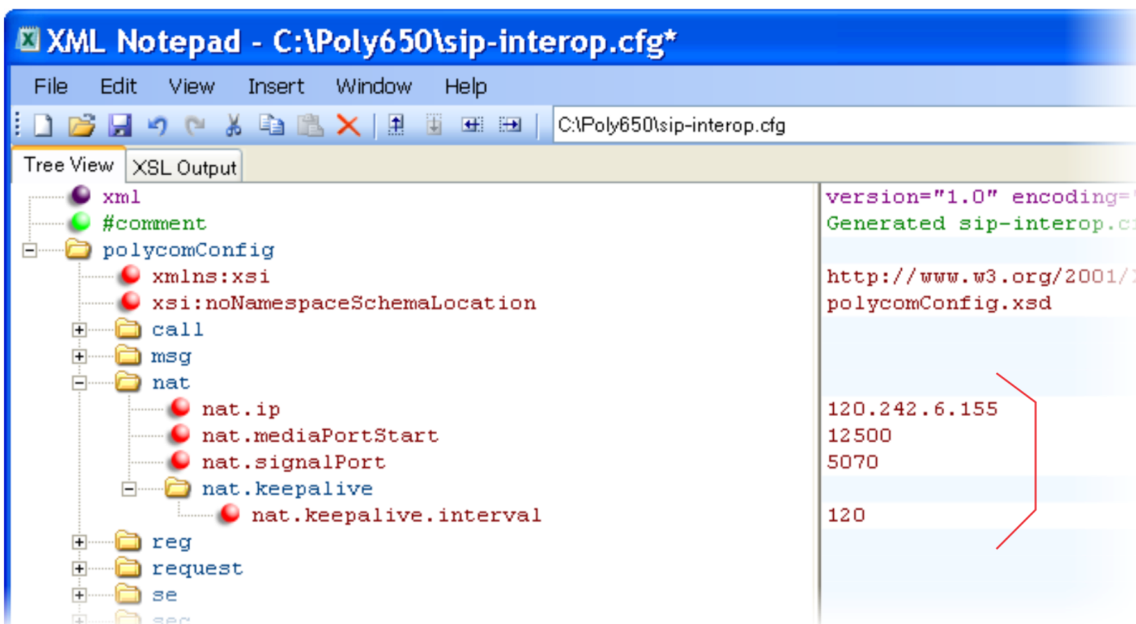
Example Network Address Translation Configuration

The following illustration shows the default NAT parameter settings. The parameter `nat.ip` is the public IP that you want to advertise in SIP signaling. The default IP is 120.242.6.155.

The parameter `nat.mediaPortStart` is the RTP used to send media. If non-Null, this attribute will set the initially allocated RTP port and will override the value set in `tcpIpApp.port.rtp.mediaPortRangeStart`. In the example, the starting port is 12500 and the phone will cycle through start-port + 47 for phones that support audio only or start-port + 95 for phones that support video.

The parameter `nat.signalPort` specifies the port that the phone will use for SIP signaling. This parameter will override `voIpProt.local.Port`. In the example below, the phone will use port 5070 for SIP traffic.

Use the `nat.keepalive.interval` to specify the keepalive interval in seconds. This parameter sets the interval at which phones will send a keepalive packet to the gateway/NAT device. The keepalive packet keeps the communication port open so that NAT can continue to function as initially set up. In the example below, the phone will send the keepalive every 120 seconds.



Use the Corporate Directory

You can connect your phone to a corporate directory server that supports the Lightweight Directory Access Protocol (LDAP) version 3. The corporate directory is a flexible feature and table Use the Corporate Directory links you to the parameters you can configure. Once set up on the phones, the corporate directory can be browsed or searched. You can call numbers and save entries you retrieve from the LDAP server to the local contact directory on the phone.

The CX5500 system currently supports the following LDAP servers:

- Microsoft® Active Directory 2003 SP2
- Sun ONE Directory Server 5.2 p6
- Open LDAP Directory Server 2.4.12
- Microsoft Active Directory Application Mode (ADAM) 1.0 SP1

The CX5500 system supports corporate directories that support server-side sorting and those that do not. For phones that do not support server-side sorting, sorting is performed on the phone.

Tip: Better Performance With Server-Side Sorting

Polycom recommends using corporate directories that have server-side sorting for better performance. Consult your LDAP Administrator when making any configuration changes for the corporate directory. For more information on LDAP attributes, see RFC 4510 - Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map.

Web Info: Supported LDAP Directories

Configuration of a corporate directory depends on the LDAP server you use. For detailed explanations and examples of all currently supported LDAP directories, see *Technical Bulletin 41137: Best Practices When Using Corporate Directory on Polycom Phones*.

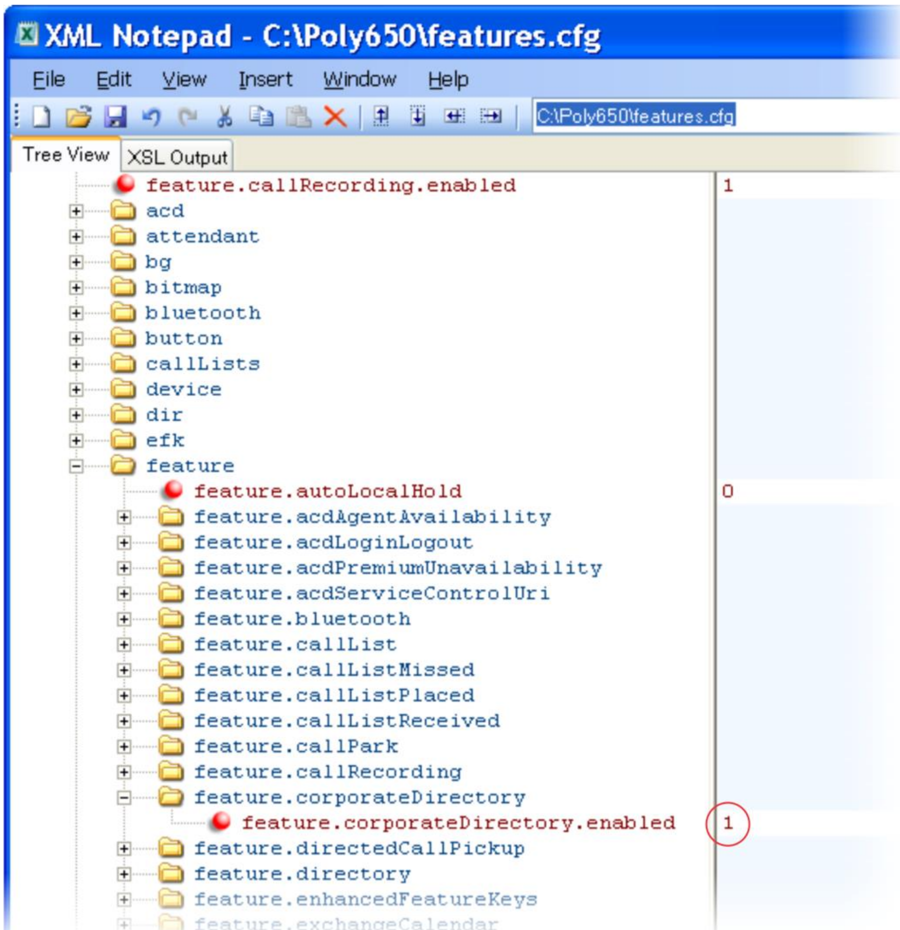
Use the Corporate Directory

Central Provisioning Server	**template** > parameter
Specify the location of the corporate directory's LDAP server, the LDAP attributes, how often to refresh the local cache from the LDAP server, and other settings	**features.cfg** > dir.corp.*
Local Phone User Interface	
Specify if the corporate directory should remember the previous search filter by navigating to **Settings > Basic > Preferences > Corporate Directory > View Persistency**. Review the corporate directory LDAP server status by navigating to **Settings > Status > CD Server Status**. To search your corporate directory, press the **Directories** key on the phone, and select **Corporate Directory**.	

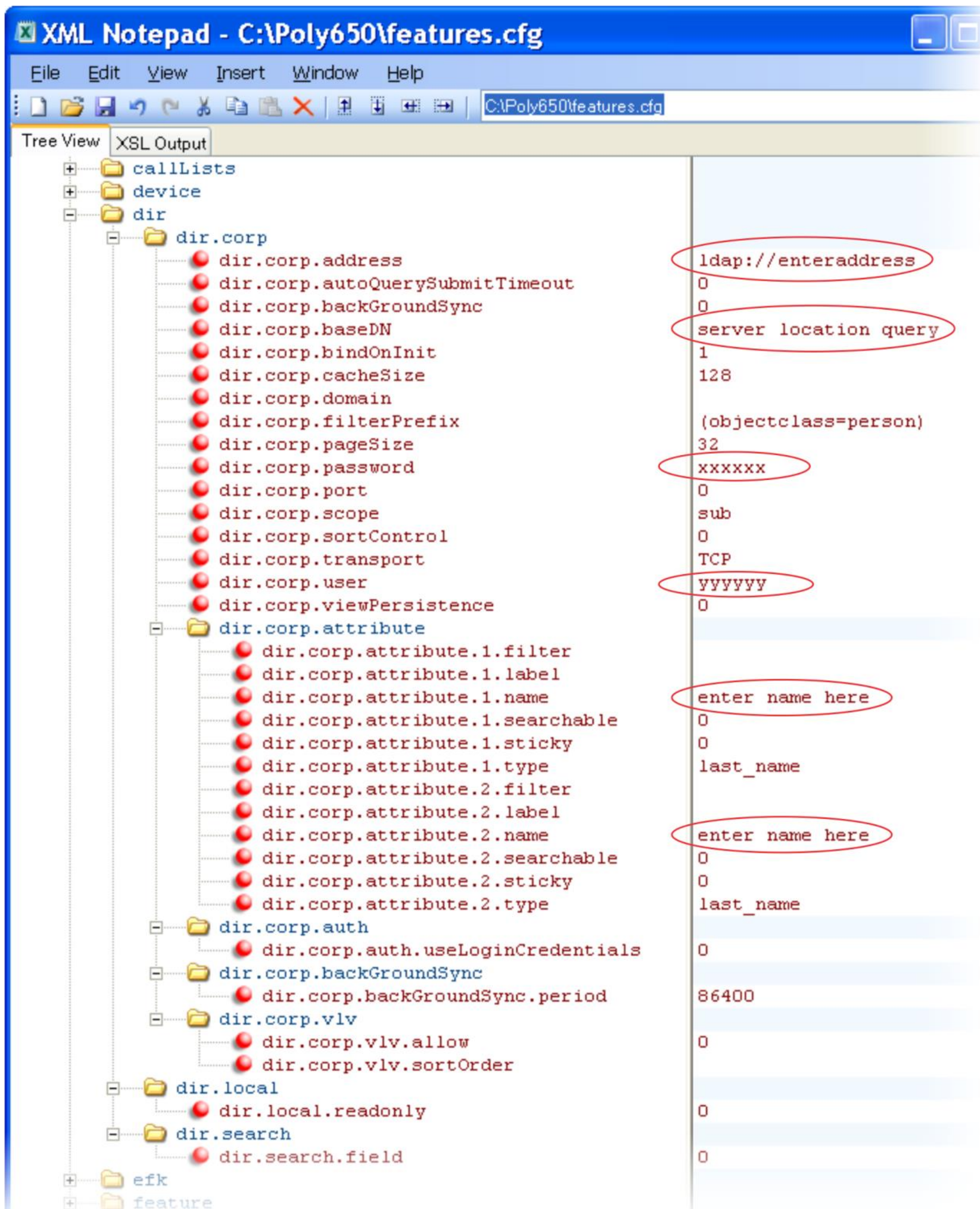
Example Corporate Directory Configuration

The following example is a representation of the minimum parameters must set to begin using the corporate directory. The exact parameters and values you will need to configure vary with the corporate directory you are using.

First, enable the corporate directory feature in the **features.cfg** template, as shown next.



The following illustration points you to the minimum parameters you need to set. Enter a corporate directory address in `dir.corp.address`, and specify where on the corporate directory server you want to make queries in `dir.corp.baseDN`. In addition, you will require a user name and password. The `dir.corp.attribute.x.name` must match the attributes in the server.



To search the corporate directory, press the **Directories** key on the phone and select **Corporate Directory**.

Configure Enhanced Feature Keys

Enhanced Feature Keys (EFK) enables you to customize the functions of a phone’s line and soft keys and, as of UC Software 4.0.1, hard keys. You can use EFK to assign frequently used functions to line keys, soft keys, and hard keys or to create menu shortcuts to frequently used phone settings.

See the table Enhanced Feature Keys for the parameters you can configure and a brief explanation of how to use the contact directory to configure line keys. Enhanced feature key functionality is implemented using star code sequences (like *69) and SIP messaging. Star code sequences that define EFK functions are written as macros that you apply to line and soft keys. The EFK macro language was designed to follow current configuration file standards and to be extensible. The macros are case sensitive.

The rules for configuring EFK for line keys, soft keys, and hard keys are different. Before using EFK, you are advised to become familiar with the macro language shown in this section and in the reference section at <efk/>.

Web Info: Using Enhanced Feature Keys

For instructions and details on how to use Enhanced Feature Keys, refer to *Feature Profile 42250: Using Enhanced Feature Keys and Configurable Soft Keys on Polycom Phones*.

Note that the configuration file changes and the enhanced feature key definitions can be included together in one configuration file. Polycom recommends creating a new configuration file in order to make configuration changes.

Tip: EFK Compatibility

The Enhanced Feature Key (EFK) feature from SIP 3.0 is compatible with Enhanced Feature Key feature from SIP 3.1. However, improvements have been made and Polycom recommends that existing configuration files be reviewed and updated.

Enhanced Feature Keys

Central Provisioning Server	template > parameter
Specify at least two calls per line key	**reg-basic.cfg** > reg.x.callsPerLineKey
Enable or disable Enhanced Feature Keys	**features.cfg** > feature.enhancedFeatureKeys.enabled
Specify the EFK List parameters	**features.cfg** > efk.efklist.x.*
Specify the EFK Prompts	**features.cfg** > efk.efkprompt.x.*
Because line keys and their functions are linked to fields in the contact directory file - **000000000000-directory.xml** (global) or <**MACaddress**>**-directory.xml** (per phone) - you must match the contact field (ct) in the directory file to the macro name field (mname) in the configuration file that contains the EFK parameters. When you enter macro names to the contact field (ct) in the directory file, add the ‘!’ prefix to the	

macro name. For more detailed information on using the contact directory, see Use the Local Contact Directory. The template directory configuration file is named **000000000000-directory~.xml.**

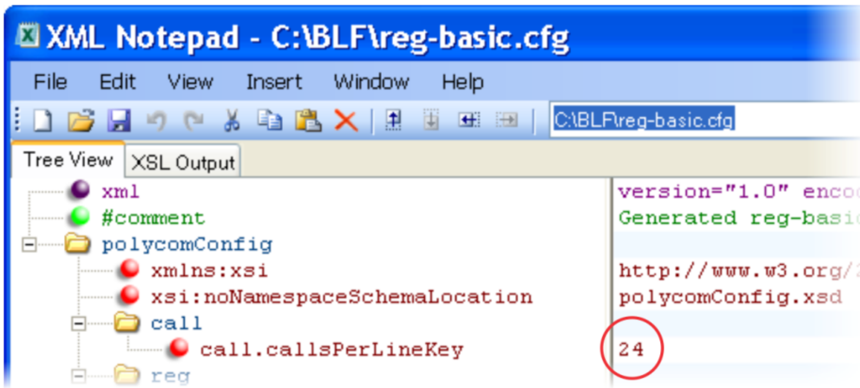
Some Guidelines for Configuring Enhanced Feature Keys

The following guidelines will help you to configure enhanced feature keys (EFKs) efficiently:

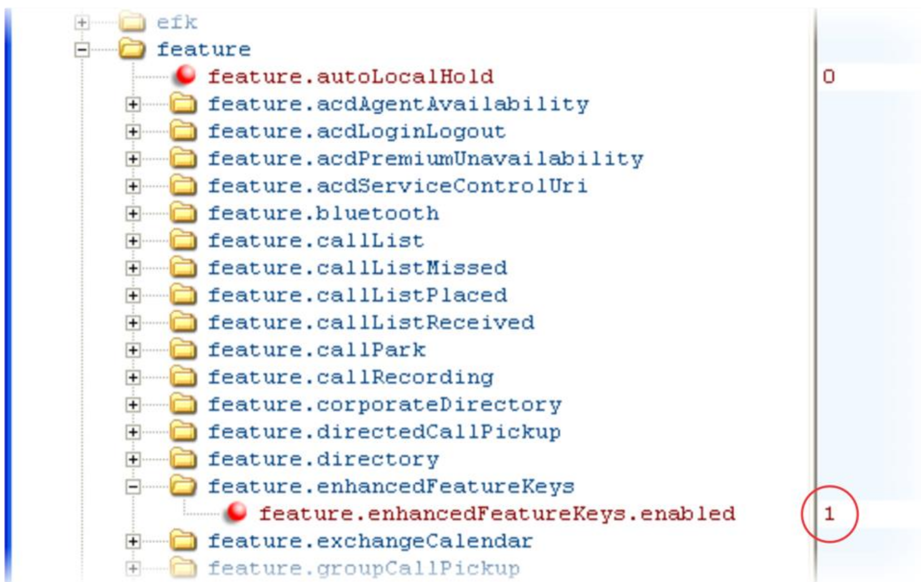
- Activation of EFK functions requires valid macro construction.
- All failures are logged at level 4 (minor).
- If two macros have the same name, the first one will be used and the subsequent ones will be ignored.
- A sequence of characters prefixed with “!” are parsed as a macro name. The exception is the speed dial reference, which starts with “!” and contains digits only.
- A sequence of characters prefixed with “^” is the action string.
- “!” and “^” macro prefixes cannot be mixed in the same macro line.
- The sequence of characters must be prefixed by either “!” or “^” so it will be processed as an enhanced feature key. All macro references and action strings added to the local directory contact field must be prefixed by either “!” or “^”.
- Action strings used in soft key definitions do not need to be prefixed by “^”. However, the “!” prefix must be used if macros or speed dials are referenced.
- A sequence of macro names in the same macro is supported (for example, “!m1!m2”).
- A sequence of speed dial references is supported (for example, “!1!2”).
- A sequence of macro names and speed dial references is supported (for example, “!m1!2!m2”).
- Macro names that appear in the local contact directory must follow the format “!<macro name>” , where <macro name> must match an <elklist> mname entry. The maximum macro length is 100 characters.
- A sequence of macros is supported, but cannot be mixed with other action types.
- Action strings that appear in the local contact directory must follow the format “^<action string>”. Action strings can reference other macros or speed dial indexes. Protection against recursive macro calls exists (the enhanced feature keys fails once you reach 50 macro substitutions).

Enhanced Feature Key Examples

The following illustration shows the default value 24 calls per line key. Ensure that you specify at least two calls per line key.

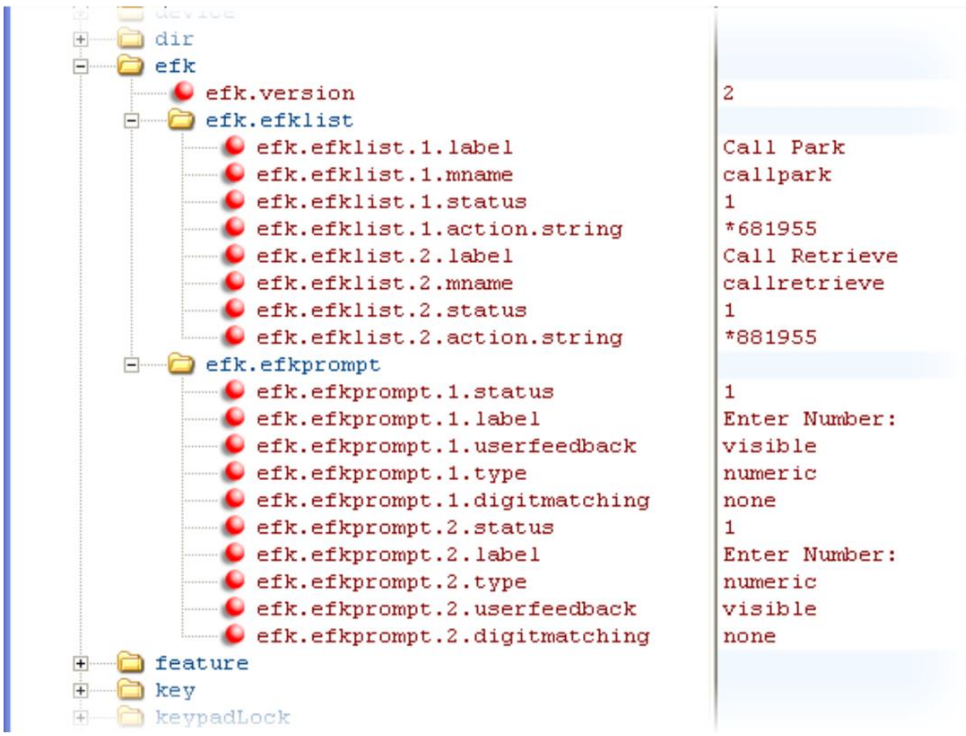


Enable the enhanced feature keys feature in the **features.cfg** template file, as shown next.



In the following illustration, the EFK parameters are located in the **features.cfg** template file. In the `efk.efklist.x.*` parameters, line key 1 has been assigned a Call Park address (1955) and line key 2 a Call Retrieve function. The parameter `acton.string` shows you the macro definition for these two functions. In addition, `status` is enabled and a label has been specified to display next to the line key. The entry in the `mname` parameter corresponds to the `contact (ct)` field in the contact directory.

In the `efk.prompt.*` parameters, `status` has been enabled. The label on the user prompt has been defined as *Enter Number:* and this prompt will display on the phone screen. The `type` parameter has been set to `numeric` to allow only numbers and because `userfeedback` has been specified as `visible`, you will be able to see the numbers you enter into the prompt.



Understanding Macro Definitions

The `efk.efklist.x.action.string` can be defined by one of the following:

- Macro Actions
- Prompt Macro Substitution
- Expanded Macros

Macro Actions

The action string is executed in the order it displays. User input is collected before any action is taken. The action string can contain the fields shown in the table Macro Actions and Descriptions.

Macro Actions and Descriptions

$L<label>$
This is the label for the entire operation. The value can be any string including the null string (in this case, no label displays). This label will be used if no other operation label collection method worked (up to the point where this field is introduced). Make this the first entry in the action string to be sure this label is used; otherwise another label may be used and this one ignored.

digits

The digits to be sent. The appearance of this parameter depends on the action string.

$C<command>$

This is the command. It can appear anywhere in the action string. Supported commands (or shortcuts) include:

hangup (`hu`)

hold (`h`)

waitconnect (`wc`)

pause <*number of seconds*> (`p` <*num sec*>) where the maximum value is 10

$T<type>$

The embedded action type. Multiple actions can be defined. Supported action types include:

`invite`

`dtmf`

`refer`

Note: Polycom recommends that you always define this field. If it is not defined, the supplied digits will be dialed using INVITE (if no active call) or DTMF (if an active call). The use of refer method is call server dependent and may require the addition of star codes.

$M<macro>$

The embedded macro. The <*macro*> string must begin with a letter. If the macro name is not defined, the execution of the action string fails.

$P<prompt num>N<num digits>$

The user input prompt string. See Prompt Macro Substitution.

$S<speed dial index>$

The speed dial index. Only digits are valid. The action is found in the `contact` field of the local directory entry pointed to by the index.

$F<internal function>$

An internal function. For more information, see Internal Key Functions.

URL

A URL. Only one per action string is supported.

Prompt Macro Substitution

The `efk.efklist.x.action.string` can be defined by a macro substitution string, **PnNn** where:

- *Pn* is the prompt *x* as defined by `efk.efkprompt.x` .
- *Nn* is the number of digits or letters that the user can enter. The value must be between 1 and 32 characters; otherwise the macro execution will fail. The user needs to press the **Enter** soft key to complete data entry.

The macros provide a generic and easy to manage way to define the prompt to be displayed to the user, the maximum number of characters that the user can input, and the action that the phone performs once all user input has been collected. The macros are case sensitive.

If a macro attempts to use a prompt that is disabled, the macro execution fails. A prompt is not required for every macro.

Expanded Macros

Expanded macros are prefixed with the ^ character and are inserted directly into the local directory `contact` field. For more information, see Use the Local Contact Directory.

Special Characters

The following special characters are used to implement the enhanced feature key functionality. Macro names and macro labels cannot contain these characters. If they do, you may experience unpredictable behavior.

- ! The characters following it are a macro name.
- ' or ASCII (0x27) This character delimits the commands within the macro.
- $ This character delimits the parts of the macro string. This character must exist in pairs, where the delimits the characters to be expanded.
- ^ This character indicates that the following characters represent the expanded macro (as in the action string).

Example Macro

The action string

`$Changup$*444*$P1N4$$Tinvite$$Cwaitconnect$$P2N3$$Cpause2$$Tdtmf$$Changup$`

is executed in order as follows:

a The user is prompted for 4 digits. For example, *1234*.

b The user is prompted for 3 digits. For example, 567.

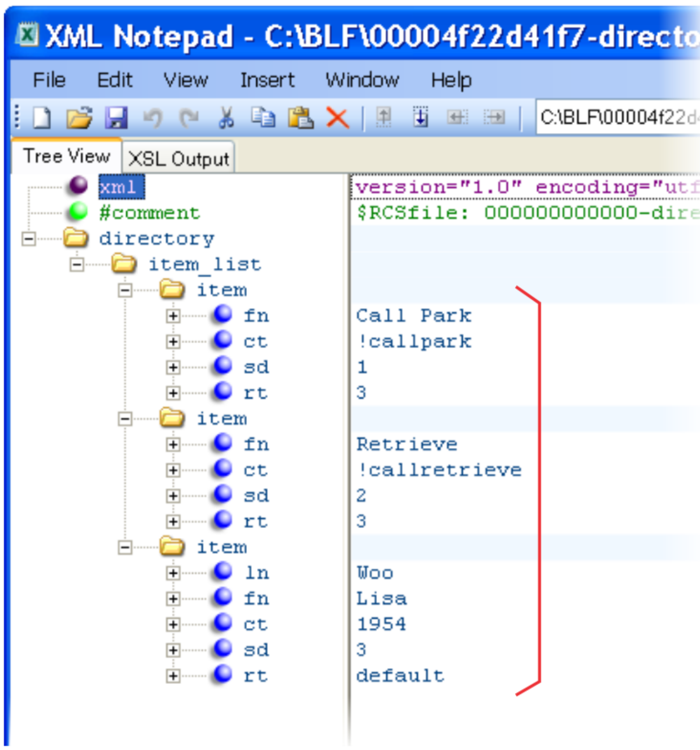
c The user's active call is disconnected.

d The string *444*1234 is sent using the INVITE method.

e Once connected, there is a 2 second pause, and then the string 567 is sent using DTMF dialing on the active call.

f The active call is disconnected.

Because line keys and their functions are linked to fields in the directory file, a macro name you enter in `efk.list.x.mname` must match the name you enter to the `contact (cn)` field in the directory file. The macro name you enter in the `(ct)` field of the directory file must begin with the '!' prefix. The following example directory file shows a line key configured with Call Park, Call Retrieve, and a speed dial contact Lisa Woo.



For an explanation of all fields in the directory file, see the table Understanding the Local Contact Directory.

Speed Dial Example

If your organization's voicemail system is accessible through 7700 and your voicemail password is 2154, you can use a speed dial key to access your voicemail by entering *7700$Cpause3$2154* as the contact number in the `contact (ct)` element.

Tip: Ensuring Users Do Not Delete Definitions in the Contact Directory

To avoid users accidentally deleting the definitions in the contact directory, make the contact directory read only.

Configure Soft Keys

You can customize the functions of the phone’s soft keys. This feature is typically used to access frequently used functions or to create menu shortcuts to frequently used phone settings. The parameters that configure soft keys are shown in the table Configure Soft Keys. As with EFK line keys, you assign functions to soft keys using macros. For a list of the available macros, see the topic Understanding Macro Definitions in the Configure Enhanced Feature Keys section.

You can configure the soft keys to display functions depending on the phone’s menu level or call state. For example, you can make a Call Park soft key available when the phone is in an active call state.

Custom soft keys can be added in the following call states:

- **Idle** There are no active calls.
- **Active** This state starts when a call is connected. It stops when the call stops or changes to another state (like hold or dial tone).
- **Alerting** (or ringing or incoming proceeding) The phone is ringing.
- **Dial tone** You can hear a dial tone.
- **Proceeding** (or outgoing proceeding) This state starts when the phone sends a request to the network. It stops when the call is connected.
- **Setup** This state starts when the user starts keying in a phone number. This state ends when the Proceeding state starts.
- **Hold** The call is put on hold locally.

You can disable the display of any default soft key to make room for custom soft keys. Or, if your phone does not have a particular hard key, you may want to create a soft key. For example, if the phone does not have a **Do Not Disturb** hard key, you can create a **Do Not Disturb** soft key.

New soft keys can be created as:

- An Enhanced Feature Key sequence
- A speed dial contact directory entry
- An Enhanced Feature Key macro
- A URL
- A chained list of actions

The default soft keys that can be disabled include:

- **New Call**
- **End Call**
- **Split**
- **Join**
- **Forward**
- **Directories**
- **MyStatus** and **Buddies**
- **Hold**, **Transfer**, and **Conference**

Note: Inserting Soft Keys Between the Hold, Transfer, and Conference Soft Keys

The **Hold**, **Transfer**, and **Conference** soft keys are grouped together to avoid usability issues. You may experience errors if you try to insert a soft key between these three grouped soft keys.

If you want your phone to display both default and custom soft keys, you can configure them in any order. However, the order in which soft keys display depends on the phone's menu level and call state. If you have configured custom soft keys to display with the default soft keys, the order of the soft keys may change.

Up to 10 custom soft keys can be configured. If more soft keys are configured than fit on the phone's screen, a **More** soft key displays. Press the **More** soft key to view the remaining soft keys.

The table Configure Soft Keys shows you the parameters for configuring soft keys. However, this feature is part of Enhanced Feature Keys (EFK) and you must enable the enhanced feature keys parameter to configure soft keys. See the section Configuring Enhanced Feature Keys for details about configuring soft keys and line keys on the phone.

Configure Soft Keys

Central Provisioning Server	**template** > parameter
To turn Enhanced Feature Keys on (required)	**features.cfg** > feature.enhancedFeatureKeys.enabled
Specify the macro for a line key or soft key function	**features.cfg** > softkey.x.action
To enable a custom soft key	**features.cfg** > softkey.x.enable
Specify the position of the soft key on the phone screen	**features.cfg** > softkey.x.insert
Specify the text to display on the soft key label	**features.cfg** > softkey.x.label
To position the custom soft key before the default soft keys	**features.cfg** > softkey.x.precede
Specify which call states the soft key will display in	**features.cfg** > softkey.x.use.*
To display soft keys for various phone features, including default soft keys	**features.cfg** > softkey.feature.*

Example Soft Key Configurations

This section provides a few examples of available soft key configurations.

Web Info: Using Configurable Soft Keys

For more examples, see *Feature Profile 42250: Using Enhanced Feature Keys and Configurable Soft Keys on Polycom Phones*.

To disable the New Call soft key:

1. In the **features.cfg** template file, set `softkey.feature.newcall` to '0'.
2. Reboot the phone.

   The **New Call** soft key is not displayed and the soft key space it occupied is empty.

To map a chained list of actions to a soft key:

1. Configure speed dial index 2 in the contact directory file with a phone address. For example, enter '2900' in the contact (ct) field.
2. In the contact directory, enter '!2' in the contact (ct) field of speed dial index 1.
3. Update the configuration file as follows:

```
softkey.1.label = ChainAct
softkey.1.action = $S1$$Tinvite$
softkey.1.use.idle = 1
```

4. Reboot the phone.

   A soft key **ChainAct** displays. Press **ChainAct** to dial the phone number 2900.

To map the Do Not Disturb Enhanced Feature Key sequence to a soft key:

1. Update the configuration file as follows:

```
softkey.1.label = DND
softkey.1.action = $FDoNotDisturb$
softkey.1.use.idle = 1
```

2. Reboot the phone.

   A **DND** soft key is displayed on the phone when it is in the idle state. When the **DND** soft key is pressed, the Do Not Disturb icon is displayed.

To map a Send-to-Voicemail Enhanced Feature Key sequence to a soft key:

1. Update the configuration file as follows:

```
softkey.2.label = ToVMail
softkey.2.action = ^*55$P1N10$$Tinvite$
softkey.2.use.alerting = 1
```

2. Reboot the phone.

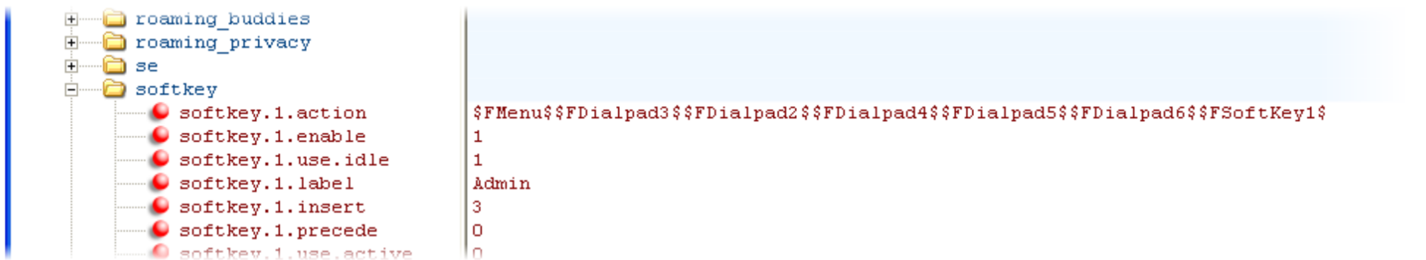
   When another party calls, the **ToVMail** soft key is displayed. When the user presses the **ToVMail** soft key, the other party is transferred to voicemail.

Tip: Active Call Transfer Star Codes Depend On Your Call Server

The exact star code to transfer the active call to Voicemail depends on your call server.

The following example enables a soft key in the phone’s idle state that navigates to a phone’s administrator settings. The soft is inserted in soft key position 3, after the default soft keys. Note the macro action string:

`$FMenu$$FDialpad3$$FDialpad2$$FDialpad4$$FDialpad5$$FDialpad6$$FSoftKey1$`



Enable the Power Saving Feature

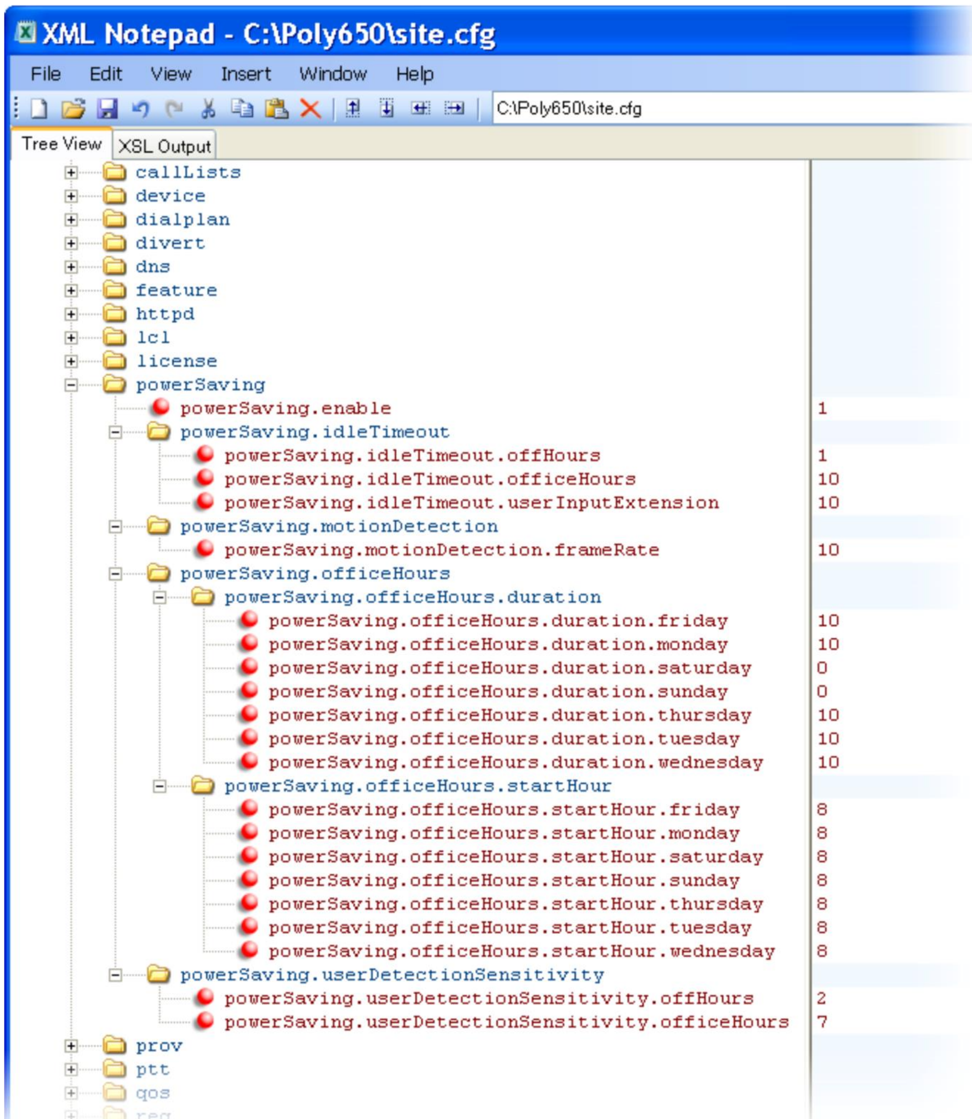
CX5500 systems support a power-saving feature, which is disabled by default. This feature has a number of options you can configure, as listed in the table Power Saving. You can turn on the phone’s power-saving feature during non-working hours and working hours. If you want to turn on power-saving during non-working hours, you can configure the power-saving feature around your work schedule. Or, if you want to turn on the power-saving feature while at work, you can configure the sensitivity of the phone’s motion detection system and an idle time after which the phone enters the power-saving mode.

Power Saving

Central Provisioning Server	**template** > parameter
Turn the power-saving feature on or off	**site.cfg** > powerSaving.enable
Specify the amount of time before the phone screen goes idle	**site.cfg** > powerSaving.idleTimeout.*
Set the office hour start time and duration for each day of the week	**site.cfg** > powerSaving.officeHours.*
Web Configuration Utility	
To turn this feature on or off and configure how it works, navigate to **Settings > Power Saving** and expand the panels to set the general, office hour, idle timeout, and user detection sensitivity settings.	
Local Phone User Interface	
To configure the Power Saving Office Hours, Timeouts, and User Detection, navigate to **Settings > Basic > Power Saving**.	

Example Power-Saving Configuration

The following illustration shows the power-saving default settings, which reflect the hours of a typical work week.



Configure Group Paging

The Group Paging feature enables you to make pages —one-way audio announcements—to users subscribed to a page `group`. Administrators must enable Paging before users can subscribe to a page group.

Paging has 25 groups you can subscribe to and announcements play only through the phone's speakerphone. To configure Group Paging, see the table Configure Group Paging.

Web Info: Using a Different IP multicast address

The Group Paging feature uses an IP multicast address. If you want to change the default IP multicast address, ensure that the new address does not already have an official purpose as specified in the IPv4 Multicast Address Space Registry.

You specify the same IP multicast address in the parameter `ptt.address` for Paging mode. Paging administrator settings shown in the table Configure Group Paging are located in the **site.cfg** template file. Page group settings are located in the **features.cfg** template file.

Configure Group Paging

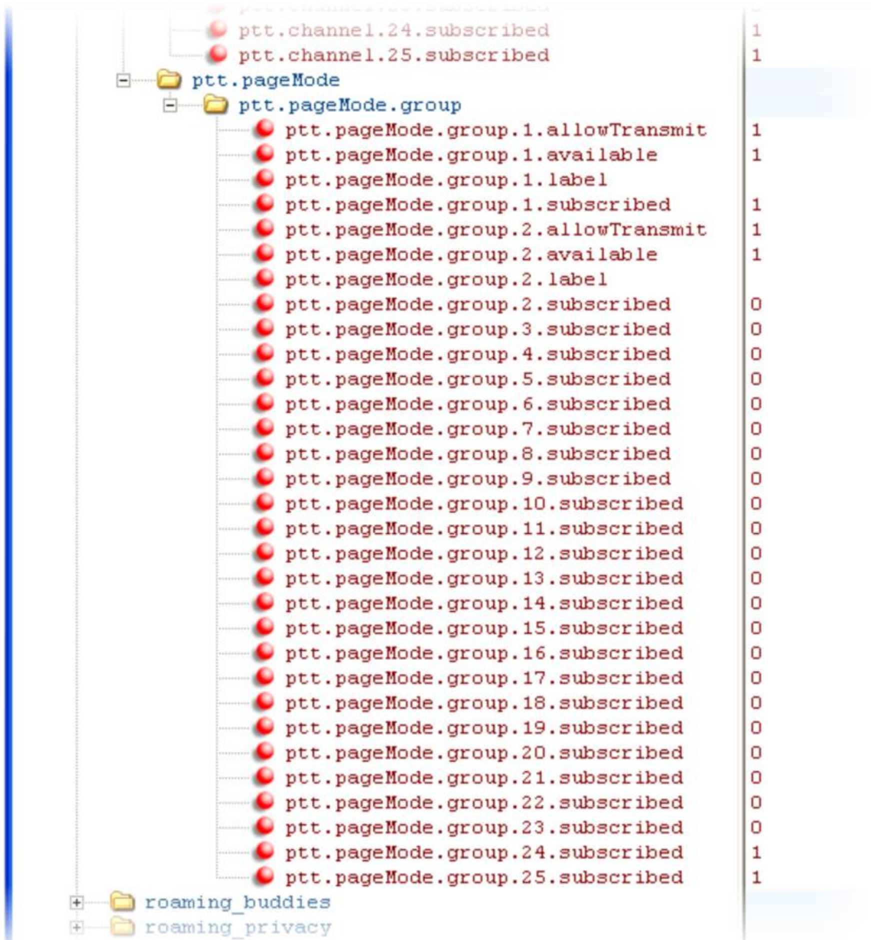
Central Provisioning Server	**template** > parameter
Specify the IP multicast address used for the paging feature	**site.cfg** > ptt.address
Enable Paging mode	**site.cfg** > ptt.pageMode.enable
Specify the display name	**site.cfg** > ptt.pageMode.displayName
Specify settings for all Page groups	**features.cfg** > ptt.pageMode.group.*
Web Configuration Utility	
To specify the IP multicast address and port, and available paging groups for Group Paging, navigate to **Settings > Paging/PTT Configuration** and expand **Settings** and **Group Paging Configuration**.	
Local Phone User Interface	
Specify the IP multicast address and port, and available paging groups for Group Paging from the Paging/PTT Configuration menu, accessible from **Settings > Advanced > Admin Settings**. Users can access basic Group Paging settings from **Settings > Basic > Preferences > Paging/PTT Configuration**.	

Web Info: Configuring Group Paging

Though the example configurations in this section will get you started, Polycom recommends that you become familiar with the following document before using the PTT or Paging features: *Feature Profile 62327: Broadcasting Audio Messages with Group Paging and Push-to-Talk*.

Paging Mode Groups

You can subscribe to the following Paging groups. Note that groups one and two are enabled by default, and that groups 24 and 25, the priority and emergency channels respectively, are also enabled by default.



Configure Shared Call Appearances

With the shared call appearance feature enabled, an active call displays simultaneously on multiple phones in a group. By default, the answering phone has sole access to the incoming call, called line seize. You can enable another phone in the group the ability to enter a conversation, called a barge in. If the answering phone places the call on hold, that call becomes available to all phones of that group. The parameters you can configure are listed in the table Configure Shared Call Appearances. All call states of a call —active, inactive, on hold—are displayed on all phones of a group.

This feature is dependent on support from a SIP call server. To enable shared call appearances on your phone, obtain a shared line address from your SIP service provider. For more details on SIP signaling with shared call appearances, see Shared Call Appearance Signaling.

Tip: Shared Call and Bridged Line Appearances Are Distinct

Shared call appearances and bridged line appearances are similar signaling methods that enable more than one phone to share the same line or registration. The method you use varies with the SIP call server you are using.

Configure Shared Call Appearances

Central Provisioning Server	**template** > parameter
Specify the shared line address	**reg-basic.cfg** > reg.x.address
Specify the line type as shared	**reg-advanced.cfg** > reg.x.type
To disable call diversion, expose auto-holds, resume with one touch, or play a tone if line-seize fails	**sip-interop.cfg** > call.shared.*
Specify standard or non-standard behavior for processing a line-seize subscription for mutual exclusion	**sip-interop.cfg** > voIpProt.SIP.specialEvent.lineSeize.nonStandard
Specify barge-in capabilities and line-seize subscription period if using per-registration servers. A shared line will subscribe to a server providing call state information	**reg-advanced.cfg** > reg.x.*
Specify per-registration whether diversion should be disabled on shared lines	**sip-interop.cfg** > divert.x.sharedDisabled

Web Configuration Utility

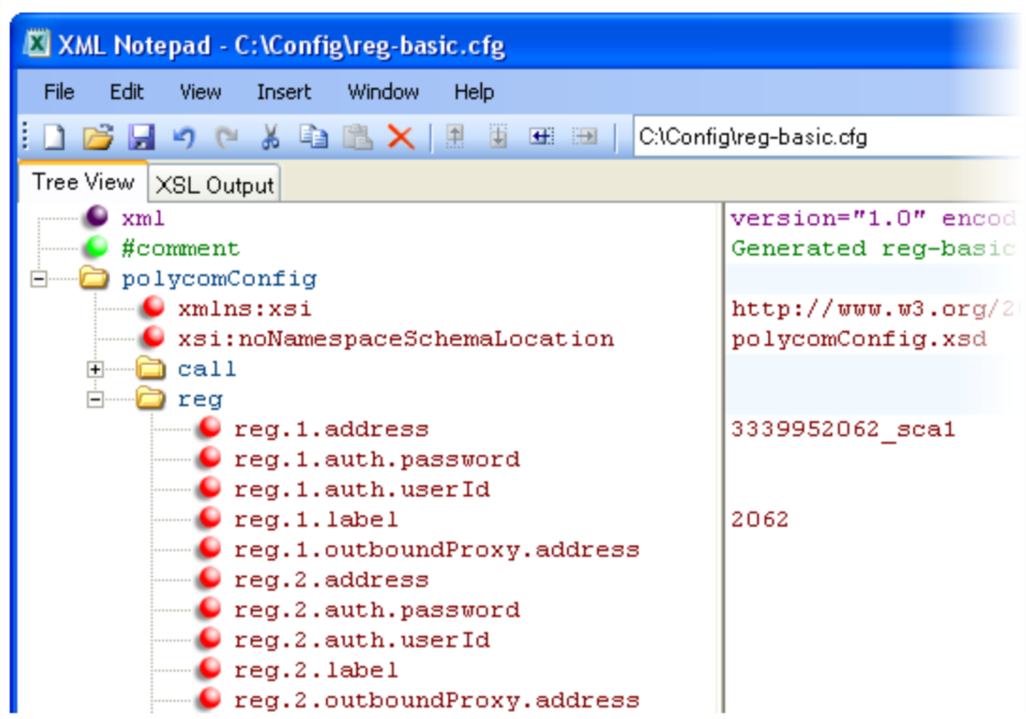
To specify the line seize subscription period for SIP Server 1 or Server 2, navigate to **Settings > SIP**, expand **Server 1** or **Server 2**, and edit the **Line Seize Timeout**.
To specify standard or non-standard behavior for processing line-seize subscription for the mutual exclusion feature, navigate to **Settings > SIP**, expand **Local Settings**, and enable or disable **Non Standard Line Seize**.
Specify the per-registration line type (shared) and the line-seize subscription behavior if you are using per-registration server, and whether diversion should be disabled on shared lines by navigating to **Settings > Lines**.

Local Phone User Interface

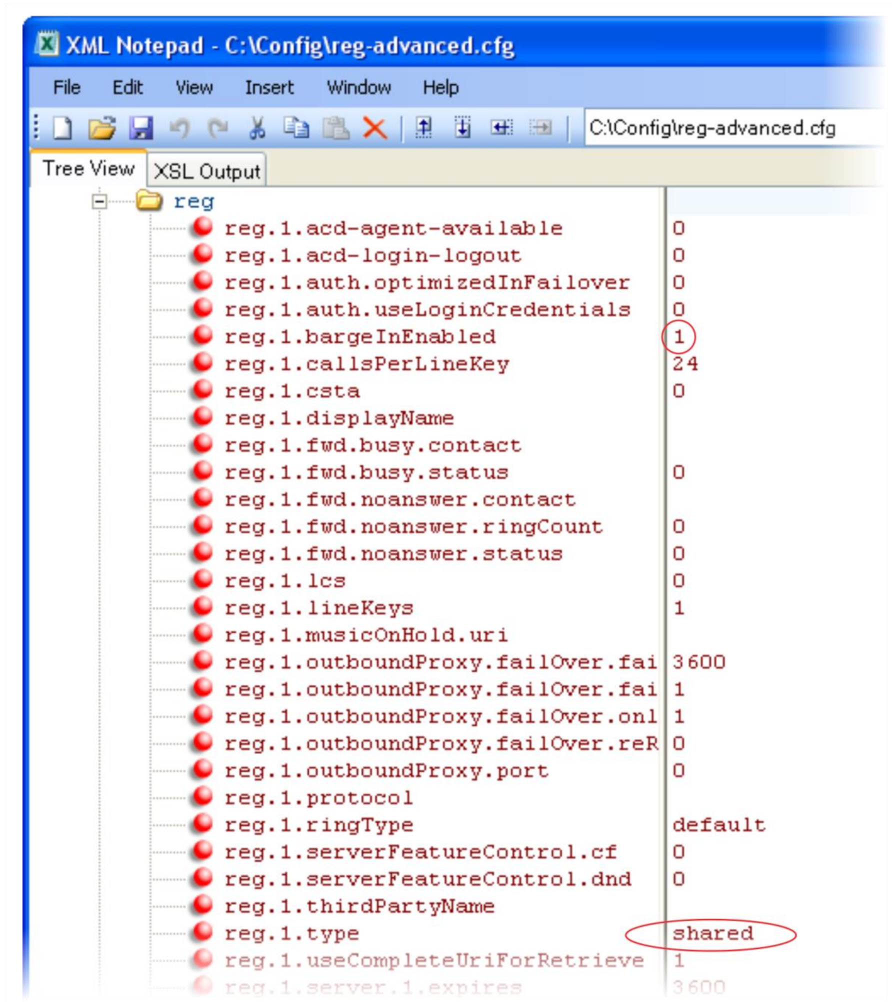
To specify the per-registration line type (shared) and shared line address, navigate to **Settings > Advanced > Admin Settings > Line Configuration > Line *X* > Line Type**.

Example Configuration

The following illustration shows the address of a registered phone line and the label that displays beside the line key, as specified in the **reg-basic.cfg** template.



If you want to configure this line to be shared, in the **reg-advanced.cfg** template, specify `shared` in `reg.1.type` . All phones that specify `shared` for registration 1 will have shared call appearance enabled for this line. In the following example, the `reg.1.bargeInEnabled` parameter is set to '1' to enable phones of this group to barge in on active calls.



After setting these parameters, activity on line 2062 displays on all phones that configure a shared call appearance for line 2062.

Enable Bridged Line Appearance

Bridged line appearance connects calls and lines to multiple phones. See the table Enable Bridged Line Appearance for a list of the parameters you can configure. With bridged line appearance enabled, an active call displays simultaneously on multiple phones in a group. By default, the answering phone has sole access to the incoming call—line seize. If the answering phone places the call on hold, that call becomes available to all phones of that group. All call states—active, inactive, on hold—are displayed on all phones of a group. For more information, see Bridged Line Appearance Signaling.

Tip: Bridged Line and Shared Call Appearances are Distinct

Shared call appearances and bridged line appearances are similar signaling methods that enable more than one phone to share the same line or registration. The methods you use vary with the SIP call server you are using. In the configuration files, bridged lines are configured by 'shared line' parameters. The barge-in feature is not available with bridged line appearances; it is available with shared call appearances.

Enable Bridged Line Appearance

Central Provisioning Server	**template** > parameter
Specify whether call diversion should be disabled by default on all shared lines	**sip-interop.cfg** > call.shared.disableDivert
Specify the per-registration line type (private or shared)	**reg-advanced.cfg** > reg.x.type
Specify the shared line third-party name	**reg-advanced.cfg** > reg.x.thirdPartyName
Specify whether call diversion should be disabled on a specific shared line (overrides default)	**reg-advanced.cfg** > divert.x.sharedDisabled
Web Configuration Utility	
To specify the line type (private or shared) and the shared line third party name for a specific line, navigate to **Settings > Lines**, choose a line from the left pane, expand Identification, and edit **Type** and **Third Party Name**. To specify whether call diversion should be disabled for a specific shared line, navigate to **Settings > Lines**, choose a line from the left pane, expand **Call Diversion**, and set **Disable Forward for Shared Lines**.	
Local Phone User Interface	
Specify the line type for each registration and the shared line third party name by navigating to **Settings > Advanced > Admin Settings > Line Configuration > Line *X***. Edit the **Line Type** and the **Third Party Name**.	

Example Bridged Line Appearance Configuration

To begin using bridged line appearance, get a registered address dedicated for use with bridged line appearance from your call server provider. This dedicated address must be assigned to a phone line in the `reg.x.address` parameter of the **reg-basic.cfg** template.

Next, in the **reg-advanced.cfg** template, enter the dedicated address in `thirdPartyName` for all phones of the BLA group and set the line type to `shared`. In this example, two or more phones can use the same dedicated address *6044533036* as the BLA address, and the line `type` has been set to `shared` from the default `private`.

For example, two phones *6044533036* and *6044533037* are configured with the 3036 BLA address. There is an incoming call to *6044533036* from *3038* that causes *3036* and *3037* phones to show the incoming call.

Enable Voicemail Integration

The phone is compatible with voicemail servers. You can configure each phone or line registration per phone to subscribe with a SIP URL to a voicemail server contact. You can also configure the phone to access voicemail with a single soft key, for example, the **Messages** icon in the status bar on the CX5500 system. When you access the voicemail server, the phone gives a visual and audio alert; you can also configure a message waiting alert to indicate that you have unread voicemail messages. The table Voicemail Integration shows you the parameters you can configure.

Voicemail Integration

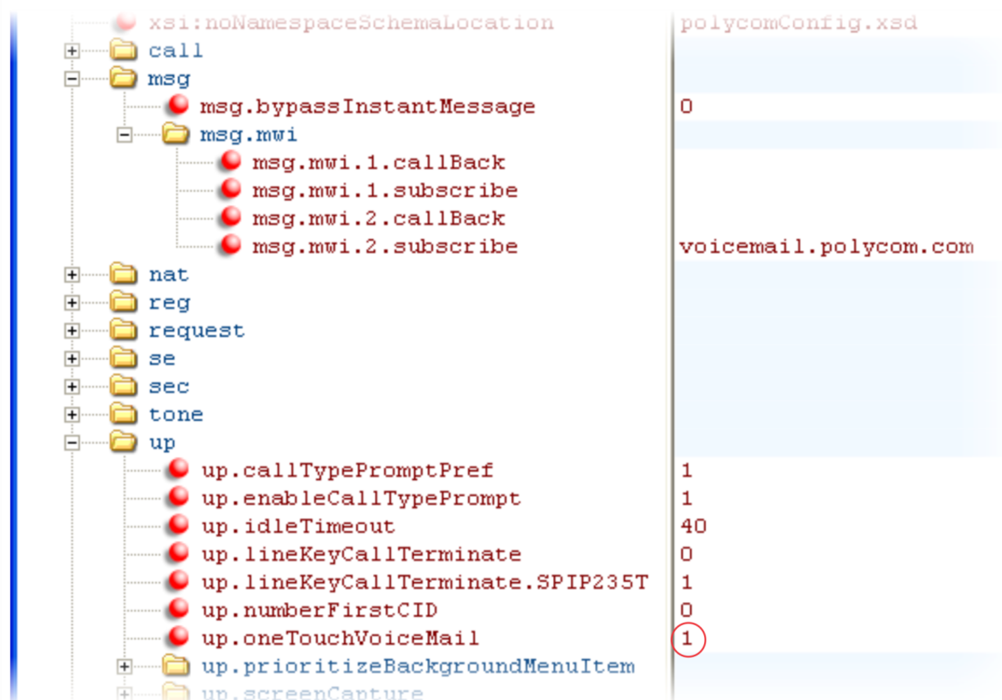
Central Provisioning Server	**template** > parameter
To turn one-touch Voicemail on or off	**sip-interop.cfg** > up.oneTouchVoiceMail
Specify the URI of the message center server	**sip-interop.cfg** > msg.mwi.x.subscribe
Set the mode of message retrieval	**sip-basic.cfg** > msg.mwi.x.callBackMode
Specify a contact number for the phone to call to retrieve messages, `callBackMode` must be set to Contact	**sip-interop.cfg** > msg.mwi.x.callBack
Specify if message waiting notifications should display or not	**site.cfg** > up.mwiVisible
Specify if the the phone screen backlight illuminates when you receive a new voicemail message	**site.cfg** > mwi.backLight.disable

Web Configuration Utility

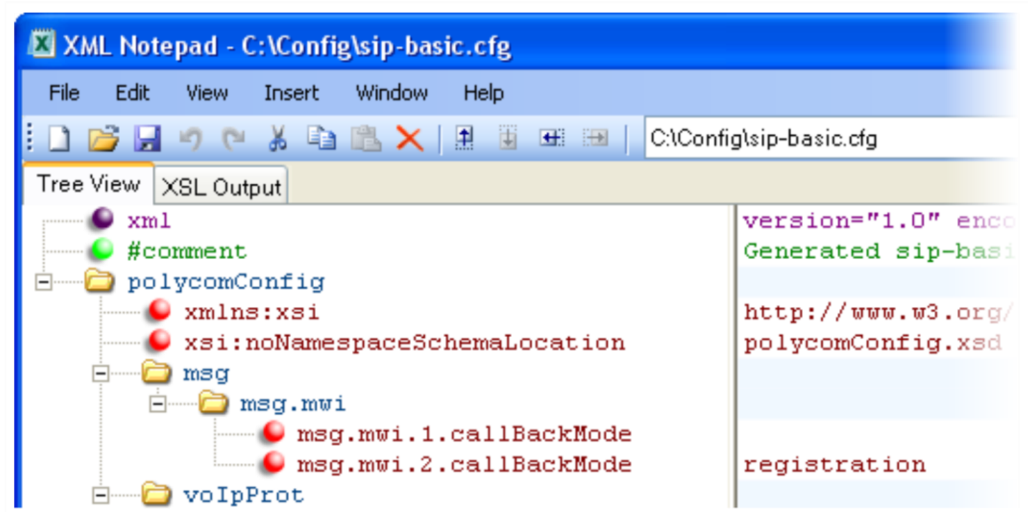
To turn One Touch Voicemail on or off, navigate to **Preferences > Additional Preferences**, expand **User Preferences**, and set **One Touch Voicemail**.
To specify the message center settings for a specific line, navigate to **Settings > Lines**, select a line from the left pane, and expand **Message Center**.

Example Voicemail Configuration

The following illustration shows you how to enable one-touch access to the voicemail server. In the next illustration, line 2 is configured to `subscribe` to the voicemail server at *voicemail.polycom.com*.



The following illustration shows that, in the **sip-basic.cfg** template, the default `callBackMode` setting for line 2 is set to `registration`. The phone will use the address assigned to line 2 to subscribe to the voicemail server you entered in `msg.mwi.2.subscribe` .



Once this is enabled in the **sip-interop.cfg** template, on the phone, press the **Messages** key and select **Message Center** to access your voicemail.

Enable Multiple Registrations

The CX5500 system can have multiple registrations; each registration requires an address, or phone number. CX5500 systems registered with Microsoft Lync Server support one Lync registration. Enable Multiple Registrations explains the registration parameters and options. The CX5500 system supports a maximum of 16 registrations.

Each registration can be mapped to one or more line keys. Note that a line key can be used for only one registration. The user can select which registration to use for outgoing calls or which to use when initiating new instant message dialogs. Note that this feature is one of several features associated with *Flexible Call Appearances*. For definitions of all features associated with Flexible Call Appearances, see the table Enable Multiple Registrations.

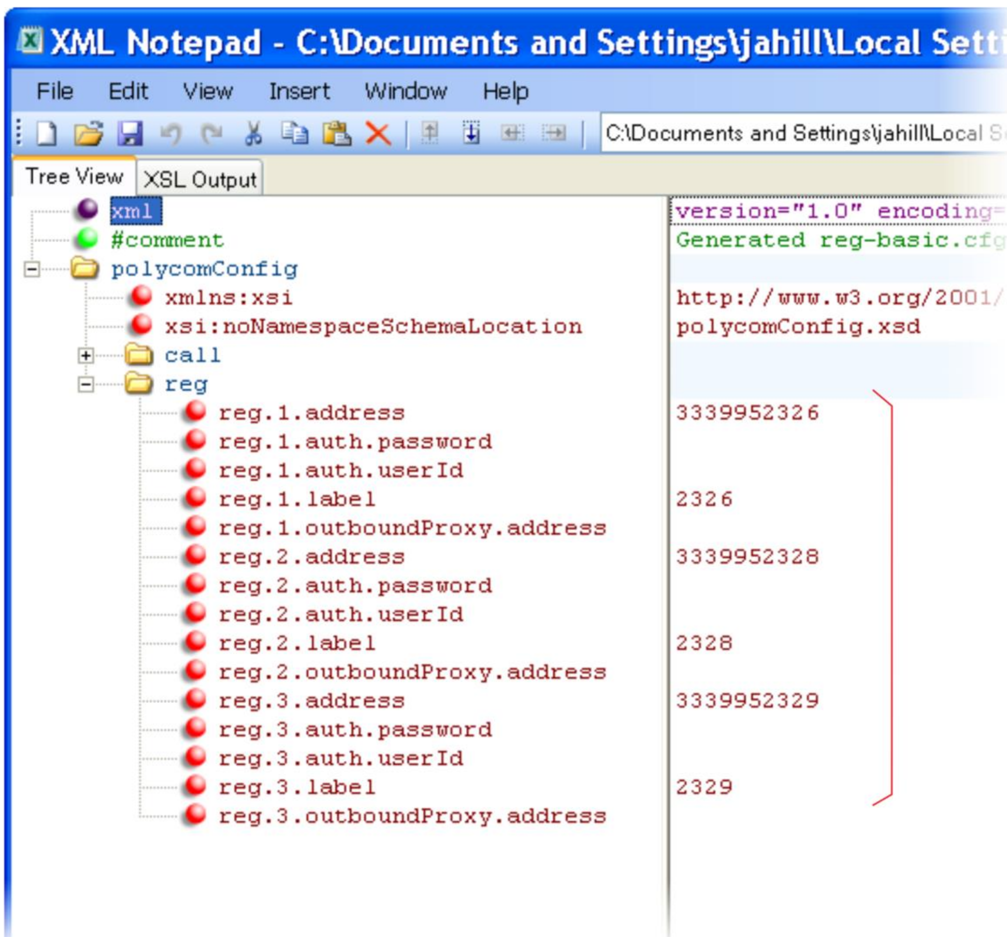
Enable Multiple Registrations

Central Provisioning Server	template > parameter
Specify the local SIP signaling port and several optional SIP servers to register to. For each server specify the registration period and the signaling failure behavior	**sip-interop.cfg** > voIpProt.SIP.* and voIpProt.server.x.*
Specify a display name, a SIP address, an optional display label, an authentication user ID and password, the number of line keys to use, and an optional array of registration servers. The authentication user ID and password are optional and for security reasons can be omitted from the configuration files. The local flash parameters will be used instead. The optional array of servers and their parameters will override the servers specified in <voIpProt.server/> if non-Null	**reg-basic.cfg**, **reg-advanced.cfg** > reg.x.*

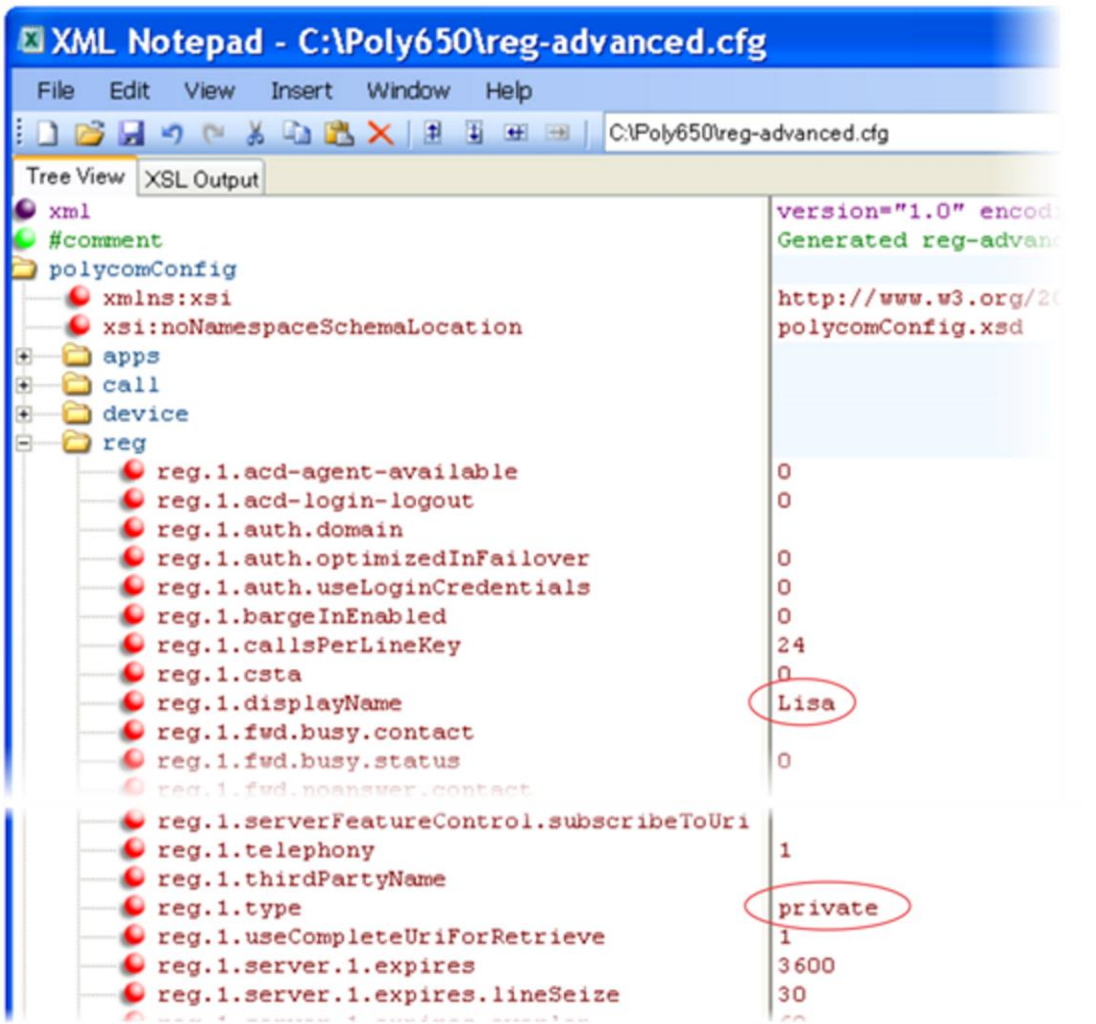
Web Configuration Utility Specify the local SIP signaling port and several optional SIP servers to register to. Specify a display name, a SIP address, an optional display label, an authentication user ID and password, the number of line keys to use, and an optional array of registration servers. The authentication user ID and password are optional and for security reasons can be omitted from the configuration files. The local flash parameters will be used instead. The optional array of servers will override the servers specified in <server/> in non-Null. Configure multiple registrations by navigating to **Settings > Lines**.
Local Phone User Interface Use the Call Server Configuration and Line Configuration menu to specify the local SIP signaling port, a default SIP server to register to, and registration information for up to twelve registrations (depending on the phone model). These configuration menus contain a sub-set of all the parameters available in the configuration files.

Example Multiple Registration Configuration

In the next illustration, in the **reg-basic.cfg** template, multiple line registrations and a label for each registration has been enabled for lines 1, 2, and 3.



In the **reg-advanced.cfg** template shown next, when you make a call using line 1, the name you enter in `reg.1.displayname` will display as your caller ID, in this case *Lisa*. The parameter `reg.x.type` is left in the default `private`, which indicates that the registration will use standard call signaling.



Set Up Server Redundancy

Server redundancy is often required in VoIP deployments to ensure continuity of phone service if, for example, where the call server needs to be taken offline for maintenance, the server fails, or the connection between the phone and the server fails. The table Set Up Server Redundancy points to several parameters you can configure.

Two types of redundancy are possible:

- **Failover** In this mode, full phone system functionality is preserved by having a second call server of equivalent capability take over from the server that went down/off-line. Use this mode of operation with DNS mechanisms or 'IP Address Moving' from the primary to the back-up server.

- **Fallback** In this mode, a second call server of lesser capability (router or gateway device) takes over call control to provide basic calling capability without some of the richer features offered by the primary call server (for example, shared lines, presence, and Message Waiting Indicator). The CX5500 system supports configuration of multiple servers per SIP registration for this purpose.

In some cases, a combination of the two may be deployed. Consult your SIP server provider for recommended methods of configuring phones and servers for failover configuration.

Note: Compatibility with Microsoft® Lync

The concurrent failover/fallback feature is not compatible with Microsoft Lync.

Caution: Old Failover Behavior Is Not Supported

Prior to SIP 2.1, the `reg.x.server.y` parameters in <reg/> could be used for failover configuration. The older behavior is no longer supported. Customers that are using the `reg.x.server.y.*` configuration parameters where y>=2 should take care to ensure that their current deployments are not adversely affected. For example, the phone will only support advanced SIP features such as shared lines, missed calls, and presence with the primary server (y=1).

Set Up Server Redundancy

Central Provisioning Server	template > parameter
Specify server redundancy options including failback mode, failback timeout, and failover registration behavior	**sip-interop.cfg** > voIpProt.server.x.failOver.*
Specify which server to contact if failover occurs	**reg-advanced.cfg** > reg.x.auth.optimizedInFailover
Override the default server redundancy options for a specific registration	**reg-advanced.cfg** > reg.x.outboundProxy.failOver.*

Web Info: Failover Configuration Details

For more information, see *Technical Bulletin 5844: SIP Server Fallback Enhancements on Polycom Phones* and *Engineering Advisory 66546: Using Optional Geographical Server Redundancy Failover Behaviors*.

DNS SIP Server Name Resolution

If a DNS name is given for a proxy/registrar address, the IP address(es) associated with that name will be discovered as specified in RFC 3263. If a port is given, the only lookup will be an A record. If no port is given, NAPTR and SRV records will be tried, before falling back on A records if NAPTR and SRV records return no results. If no port is given, and none is found through DNS, 5060 will be used. If the registration type is Transport Layer Security (TLS), 5061 will be used as the port number. See RFC 3263 for an example.

Caution: No DNS Resolution Will Cause Failover

Failure to resolve a DNS name is treated as signaling failure that will cause a failover.

Behavior When the Primary Server Connection Fails

For Outgoing Calls (INVITE Fallback)

When the user initiates a call, the phone will go through the following steps to connect the call:

1. The phone will try to call the working server.
2. If the working server does not respond correctly to the INVITE, the phone will try and make a call using the next server in the list (even if there is no current registration with these servers). This could be the case if the Internet connection has gone down, but the registration to the working server has not yet expired.
3. If the second server is also unavailable, the phone will try all possible servers (even those not currently registered) until it either succeeds in making a call or exhausts the list at which point the call will fail.

At the start of a call, server availability is determined by SIP signaling failure. SIP signaling failure depends on the SIP protocol being used:

- If TCP is used, then the signaling fails if the connection fails or the Send fails.
  - If UDP is used, then the signaling fails if ICMP is detected or if the signal times out. If the signaling has been attempted through all servers in the list and this is the last server, then the signaling fails after the complete UDP timeout defined in RFC 3261. If it is not the last server in the list, the maximum number of retries using the configurable retry timeout is used. For more information, see <server/> and <reg/>.

Caution: Use Long TTLs to Avoid DNS Timeout Delays

If DNS is used to resolve the address for Servers, the DNS server is unavailable, and the TTL for the DNS records has expired, the phone will attempt to contact the DNS server to resolve the address of all servers in its list *before* initiating a call. These attempts will timeout, but the timeout mechanism can cause long delays (for example, two minutes) before the phone call proceeds using the working server. To prevent this issue, long TTLs should be used. Polycom recommends deploying an on-site DNS server as part of the redundancy solution.

Phone Configuration

The phones at the customer site are configured as follows:

- Server 1 (the primary server) will be configured with the address of the service provider call server. The IP address of the server(s) will be provided by the DNS server, for example: `reg.1.server.1.address=voipserver.serviceprovider.com` .

- Server 2 (the fallback server) will be configured to the address of the router/gateway that provides the fallback telephony support and is on-site, for example: `reg.1.server.2.address=172.23.0.1` .

Note: Caution When Using Multiple Servers Per Registration

It is possible to configure the phone for more than two servers per registration, but you need to exercise caution when doing this to ensure that the phone and network load generated by registration refresh of multiple registrations does not become excessive. This would be of particular concern if a phone had multiple registrations with multiple servers per registration and it is expected that some of these servers will be unavailable.

Phone Operation for Registration

After the phone has booted up, it will register to all the servers that are configured.

Server 1 is the primary server and supports greater SIP functionality than other servers. For example, SUBSCRIBE/NOTIFY services used for features such as shared lines and presence, will be established only with Server 1.

Upon the registration timer expiry of each server registration, the phone will attempt to re-register. If this is unsuccessful, normal SIP re-registration behavior (typically at intervals of 30 to 60 seconds) will proceed and continue until the registration is successful (for example, when the Internet link is once again operational). While the primary server registration is unavailable, the next highest priority server in the list will serve as the working server. As soon as the primary server registration succeeds, it will return to being the working server.

Note: Failover to Servers that are Not Registered

If `reg.x.server.y.register` is set to 0, the phone will not register to that server. However, the INVITE will fail over to that server if all higher priority servers are down.

Recommended Practices for Fallback Deployments

In situations where server redundancy for fallback purpose is used, the following measures should be taken to optimize the solution:

- Deploy an on-site DNS server to avoid long call initiation delays that can result if the DNS server records expire.
- Do not use OutBoundProxy configurations on the phone if the OutBoundProxy could be unreachable when the fallback occurs. If Server 2 is not accessible through the configured proxy, call signaling with Server 2 will fail.
- Avoid using too many servers as part of the redundancy configuration as each registration will generate more traffic.
- Educate users as to the features that will not be available when in fallback operating mode.

Note: Compatibility with Microsoft® Lync

The concurrent/registration failover/fallback feature is not compatible with Microsoft® Lync.

Use the Presence Feature

The presence feature enables you to monitor the status of other remote users and phones. By adding remote users to your Buddy List, you can monitor changes in the status of remote users in real time or you can monitor remote users as speed-dial contacts. You can also manually specify your status in order to override or mask automatic status updates to others and you can receive notifications when the status of your a remote line changes. The table Use the Presence Feature lists the parameters you can configure. Note that other phone users can block you from monitoring their phones.

For more information about the Lync presence feature, see *Feature Profile 84538: Using Polycom® VVX® Business Media Phones with Microsoft® Lync™ Server 2013*.

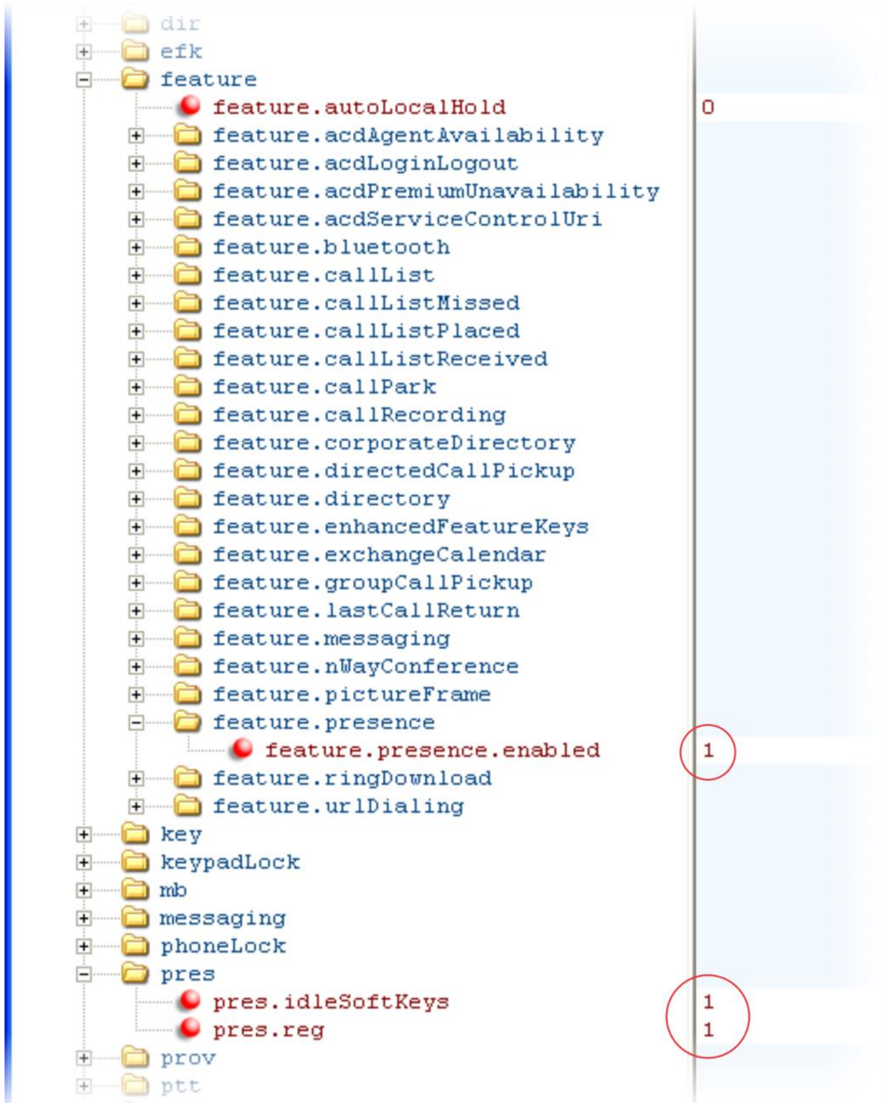
For more information about the BroadSoft UC-One presence feature, see *Feature Profile 84393: Using the Polycom® BroadSoft UC-One Application on Polycom® VVX® Business Media Phones*.

Use the Presence Feature

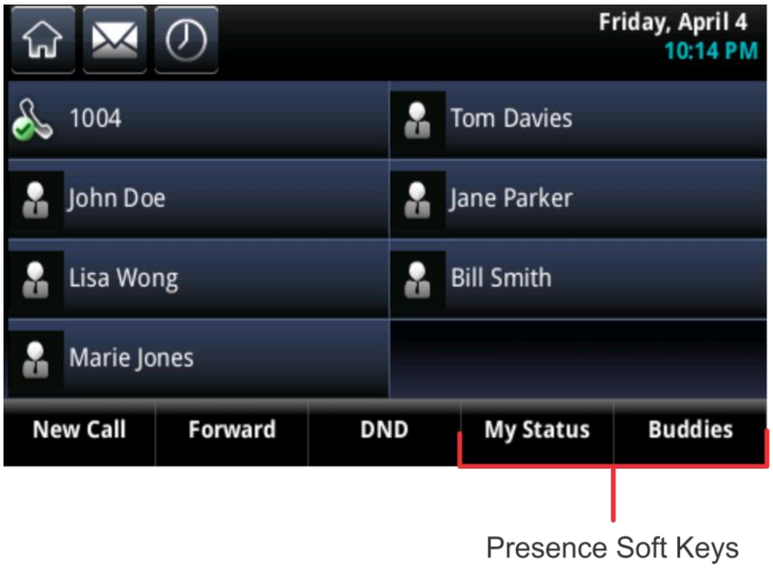
Central Provisioning Server	**template** > parameter
Specify the line/registration number used to send SUBSCRIBE for presence	**features.cfg** > pres.reg
Specify if the MyStatus and Buddies soft keys display on the Home screen	**features.cfg** > pres.idleSoftkeys
Turn the presence feature on or off	**features.cfg** > feature.presence.enabled
Local Phone User Interface	
The user can edit the directory contents. The *Buddy Watch* and *Buddy Block* fields control the buddy behavior of contacts.	

Example Presence Configuration

In the following illustration, the presence feature has been enabled in `feature.presence.enabled`. The **MyStatus** and **Buddies** soft keys will both display on the phone's home screen when you enable the `pres.idleSoftkeys` parameter. The `pres.reg` parameter will use the address of phone line 1 for the presence feature.



This configuration enables the presence feature and display the **MyStatus** and **Buddies** soft keys on the phone. When you press the **Buddies** soft key, contacts you have entered to your Buddy List display.



Configuring the Static DNS Cache

Beginning SIP 2.1.0, failover redundancy can only be used when the configured IP server hostname resolves (through SRV or A record) to multiple IP addresses. Unfortunately, the DNS cache cannot always be configured to take advantage of failover redundancy.

The solution in SIP 3.1 is to enable you to statically configure a set of DNS NAPTR SRV and/or A records into the phone. See the table Configuring the Static DNS Cache for configurable parameters.

When a phone is configured with a DNS server, it will behave as follows by default:

- The phone will make an initial attempt to resolve a hostname that is within the static DNS cache. For example, a query will be made to the DNS if the phone registers with its SIP registrar.
- If the initial DNS query returns no results for the hostname or cannot be contacted, then the values in the static cache are used for their configured time interval.
- After the configured time interval has elapsed, a resolution attempt of the hostname will again result in a query to the DNS.
- If a DNS query for a hostname that is in the static cache returns a result, the values from the DNS are used and the statically cached values are ignored.

When a phone is not configured with a DNS server, it will behave as follows:

- When the phone attempts to resolve a hostname within the static DNS cache, it will always return the results from the static cache.

Support for negative DNS caching as described in RFC 2308 is also provided to allow faster failover when prior DNS queries have returned no results from the DNS server. For more information, see RFC 2308.

Configuring the Static DNS Cache

Central Provisioning Server	**template** > parameter
Specify the line registration	**sip_interop.cfg** > reg.x.address
Specify the call server used for this registration	**sip_interop.cfg** > reg.x.server.y.*
Specify the DNS A address, hostname, and cache time interval (ttl)	**site.cfg** > dns.cache.A.x.*
Specify the DNS NAPTR parameters, including: name, order, preference, regexp, replacement, service, and ttl	**site.cfg** > dns.cache.NAPTR.x.*
Specify DNS SRV parameters, including: name, port, priority, target, ttl, and weight	**site.cfg** > dns.cache.SRV.x.*

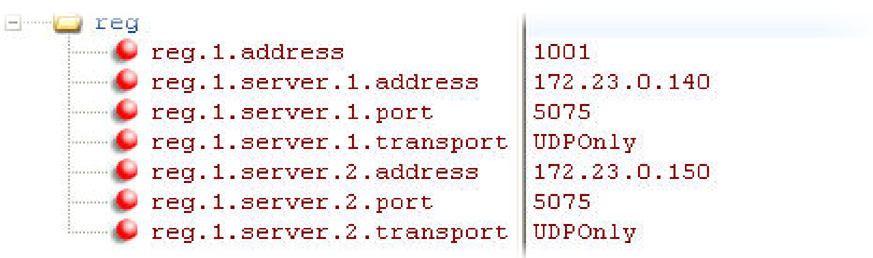
Example Static DNS Cache Configuration

The following examples show you how to configure the static DNS cache.

Example 1

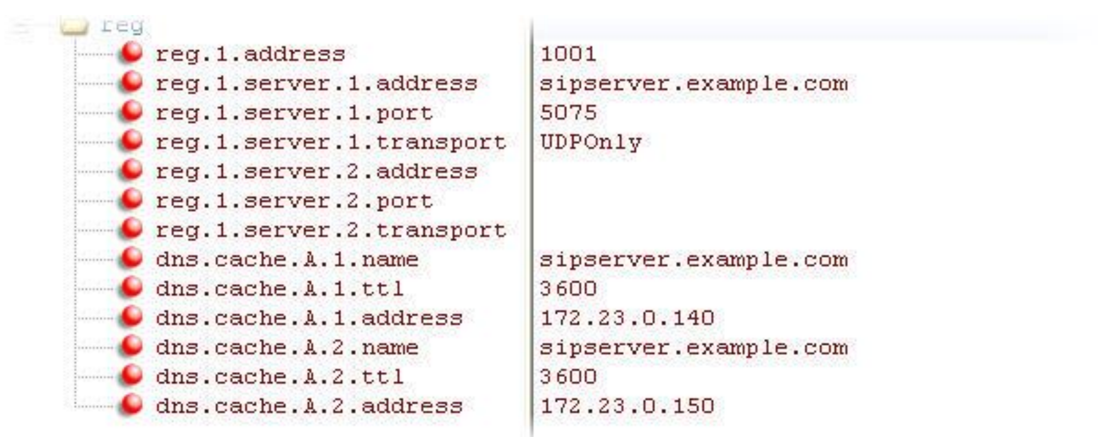
This example shows how to configure static DNS cache using A records IP addresses in SIP server address fields.

When the static DNS cache is not used, the **site.cfg** configuration will look as follows:



reg	
reg.1.address	1001
reg.1.server.1.address	172.23.0.140
reg.1.server.1.port	5075
reg.1.server.1.transport	UDPOnly
reg.1.server.2.address	172.23.0.150
reg.1.server.2.port	5075
reg.1.server.2.transport	UDPOnly

When the static DNS cache is used, the **site.cfg** configuration will look as follows:



reg	
reg.1.address	1001
reg.1.server.1.address	sipserver.example.com
reg.1.server.1.port	5075
reg.1.server.1.transport	UDPOnly
reg.1.server.2.address	
reg.1.server.2.port	
reg.1.server.2.transport	
dns.cache.A.1.name	sipserver.example.com
dns.cache.A.1.ttl	3600
dns.cache.A.1.address	172.23.0.140
dns.cache.A.2.name	sipserver.example.com
dns.cache.A.2.ttl	3600
dns.cache.A.2.address	172.23.0.150

Note: Details of the Preceding Example

Above addresses are presented to Polycom UC Software in order, for example, `dns.cache.A.1`, `dns.cache.A.2`, and so on.

Example 2

This example shows how to configure static DNS cache where your DNS provides A records for `reg.x.server.x.address` but not SRV. In this case, the static DNS cache on the phone provides SRV records. For more information, see RFC 3263.

When the static DNS cache is not used, the **site.cfg** configuration will look as follows:

reg	
reg.1.address	1002@sipserver.example.com
reg.1.server.1.address	primary.sipserver.example.com
reg.1.server.1.port	5075
reg.1.server.1.transport	UDPOnly
reg.1.server.2.address	secondary.sipserver.example.com
reg.1.server.2.port	5075
reg.1.server.2.transport	UDPOnly

When the static DNS cache is used, the **site.cfg** configuration will look as follows:

reg	
reg.1.address	1002
reg.1.server.1.address	sipserver.example.com
reg.1.server.1.port	
reg.1.server.1.transport	UDPOnly
reg.1.server.2.address	
reg.1.server.2.port	
reg.1.server.2.transport	
dns.cache.SRV.1.name	_sip._udp.sipserver.example.com
dns.cache.SRV.1.ttl	3600
dns.cache.SRV.1.priority	1
dns.cache.SRV.1.weight	1
dns.cache.SRV.1.port	5075
dns.cache.SRV.1.target	primary.sipserver.example.com
dns.cache.SRV.2.name	_sip._udp.sipserver.example.com
dns.cache.SRV.2.ttl	3600
dns.cache.SRV.2.priority	2
dns.cache.SRV.2.weight	1
dns.cache.SRV.2.port	5075
dns.cache.SRV.2.target	secondary.sipserver.example.com

Settings: Port Value Settings

The `reg.1.server.1.port` and `reg.1.server.2.port` values in this example are set to null to force SRV lookups.

Example 3

This example shows how to configure static DNS cache where your DNS provides NAPTR and SRV records for `reg.x.server.x.address`.

When the static DNS cache is used, the **site.cfg** configuration will look as follows:

reg	
reg.1.address	1002@sipserver.example.com
reg.1.server.1.address	172.23.0.140
reg.1.server.1.port	5075
reg.1.server.1.transport	UDPOnly
reg.1.server.2.address	172.23.0.150
reg.1.server.2.port	5075
reg.1.server.2.transport	UDPOnly

reg	
reg.1.address	1002@sipserver.example.com
reg.1.server.1.address	172.23.0.140
reg.1.server.1.port	5075
reg.1.server.1.transport	UDPOnly
reg.1.server.2.address	172.23.0.150
reg.1.server.2.port	5075
reg.1.server.2.transport	UDPOnly

When the static DNS cache is used, the **site.cfg** configuration will look as follows:

reg	
reg.1.address	1002
reg.1.server.1.address	sipserver.example.com
reg.1.server.1.port	
reg.1.server.1.transport	
reg.1.server.2.address	
reg.1.server.2.port	
reg.1.server.2.transport	
dns.cache.NAPTR.1.name	sipserver.example.com
dns.cache.NAPTR.1.ttl	3600
dns.cache.NAPTR.1.order	1
dns.cache.NAPTR.1.preference	1
dns.cache.NAPTR.1.flag	s
dns.cache.NAPTR.1.service	SIP+D2U
dns.cache.NAPTR.1.regexp	
dns.cache.NAPTR.1.replacement	_sip._udp.sipserver.example.com
dns.cache.SRV.1.name	_sip._udp.sipserver.example.com
dns.cache.SRV.1.ttl	3600
dns.cache.SRV.1.priority	1
dns.cache.SRV.1.weight	1
dns.cache.SRV.1.port	5075
dns.cache.SRV.1.target	primary.sipserver.example.com
dns.cache.SRV.2.name	_sip._udp.sipserver.example.com
dns.cache.SRV.2.ttl	3600
dns.cache.SRV.2.priority	2
dns.cache.SRV.2.weight	1
dns.cache.SRV.2.port	5075
dns.cache.SRV.2.target	secondary.sipserver.example.com
dns.cache.A.1.name	primary.sipserver.example.com
dns.cache.A.1.ttl	3600
dns.cache.A.1.address	172.23.0.140
dns.cache.A.2.name	secondary.sipserver.example.com
dns.cache.A.2.ttl	3600
dns.cache.A.2.address	172.23.0.150

Settings: Forcing NAPTR Lookups

The `reg.1.server.1.port`, `reg.1.server.2.port`, `reg.1.server.1.transport`, and `reg.1.server.2.transport` values in this example are set to null to force NAPTR lookups.

Web Info: Using a Static DNS Cache

For more information about using a static DNS cache, see *Technical Bulletin 36033: Using a Static DNS Cache with SoundPoint IP and SoundStation IP Phones*.

Displaying SIP Header Warnings

The warning field from a SIP header may be configured to display a three second pop-up message on the phone, for example, that a call transfer failed due to an invalid extension number. For more information, see Header Support.

You can display these pop-up messages in any language supported by the phone. The messages will display for three seconds unless overridden by another message or action. To turn the warning display on or off or specify which warnings are displayable, you can configure the parameters in Displaying SIP Header Warnings.

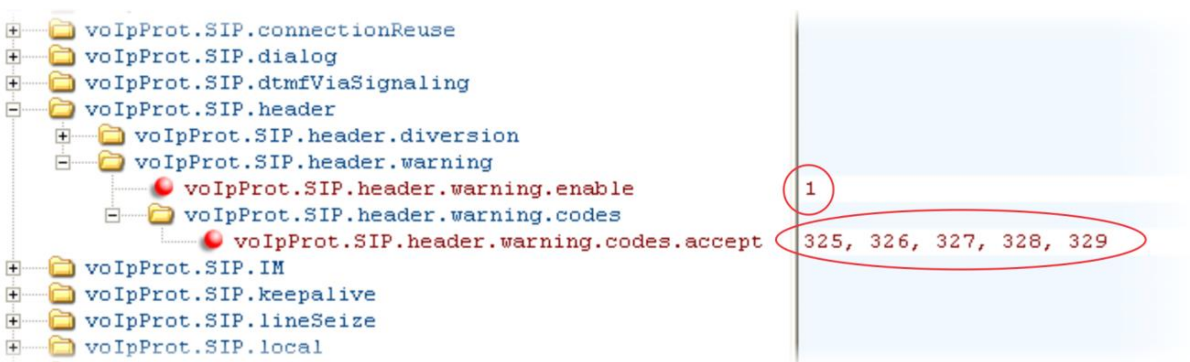
Displaying SIP Header Warnings

Central Provisioning Server	**template** > parameter
Turn this feature on or off	**sip-interop.cfg** > voIpProt.SIP.header.warning.enable
Specify which warnings are displayable	**sip-interop.cfg** > voIpProt.SIP.header.warning.codes.accept

Example Display of Warnings from SIP Headers Configuration

To enable the display of warnings from SIP headers, set the `voIpProt.SIP.header.warning.enable` parameter in the **features.cfg** template to 1. Enter the warning codes as a comma-separated string. The strings associated with the values 325 to 329 that display on the phone screen, as shown in the next illustration, have been entered automatically by the call server and are not entered by the administrator in the configuration file.

The following illustration shows a sample configuration from the **sip-interop.cfg** template file:



Quick Setup of the CX5500 System

A Quick Setup feature was added to simplify the process of entering the provisioning (boot) server parameters from the phone's user interface. This feature is designed to make it easier for on-site *out of the box* provisioning of the CX5500 system.

When you enable this feature, a **QSetup** soft key will display on the phone. When you press the **QSetup** soft key, a new menu will display. The menu enables you to access the provisioning server and quickly configure the phone to work. After configuring the Quick Setup, you can disable display of the **QSetup** soft key using a configuration file setting. The table Quick Setup of the CX5500 System indicates the parameter that enables this feature.

You can enable the Quick Setup feature through the **site.cfg** configuration file or through the phone's menu.

Web Info: Configuring Quick Setup

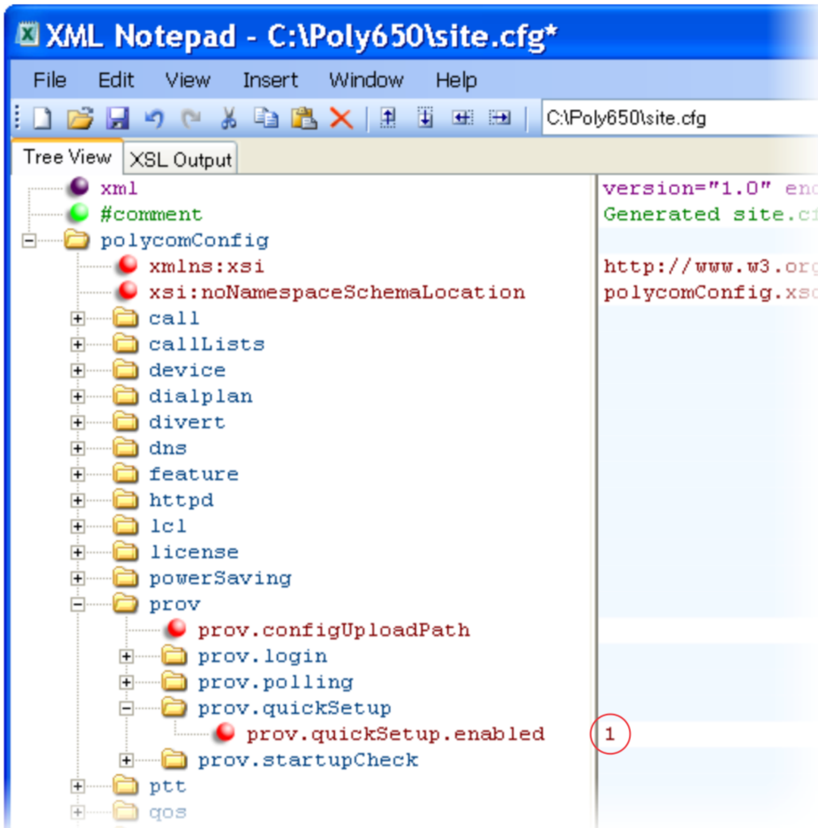
For details on how to configure quick setup, see *Technical Bulletin 45460: Using Quick Setup with Polycom Phones*.

Quick Setup of the CX5500 System

Central Provisioning Server	**template** > parameter
To enable or disable Quick Setup	**site.cfg** > prov.quickSetup.enabled

Example Quick Setup Configuration

To enable the Quick Setup feature, enable the `prov.quickSetup.enabled` parameter in the **site.cfg** template file, shown next.



The **QSetup** will display on the phone screen. Press the **QSetup** soft key to open the menu and access the quick setup feature.

Provisional Polling of the CX5500 System

You can configure how your phone provisioning automatically by configuring the parameters in the table Provisional Polling of the CX5500 System.

You can set the phone's automatic provisioning behavior to be:

- **Absolute** The phone polls at the same time every day.
- **Relative** The phone polls every x seconds, where x is a number greater than 3600.
- **Random** The phone polls randomly based on a time interval you set.
  - If the time period is less than or equal to one day, the first poll is at a random time, x, between the phone starting up and the polling period. Afterwards, the phone will poll every x seconds.

- ○ If you set the polling period to be greater than one day with the period rounded up to the nearest day, the phone polls on a random day based on the phone's MAC address, and within a random time set by the start and end polling time.

For example:

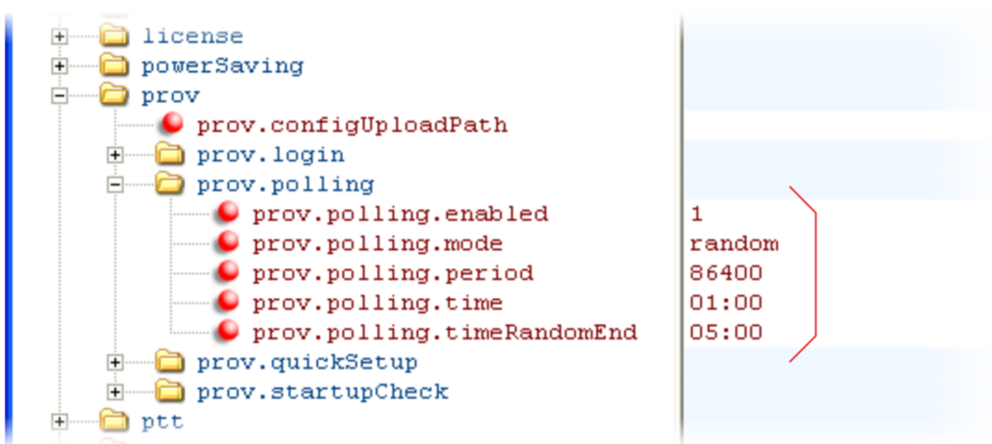
- If `prov.polling.mode` is set to `rel` and `prov.polling.period` is set to *7200*, the phone polls every two hours.
- If `prov.polling.mode` is set to `abs` and `prov.polling.timeRandomEnd` is set to *04:00*, the phone polls at 4am every day.
- If `prov.polling.mode` is set to `random`, `prov.polling.period` is set to *604800 (7 days)*, `prov.polling.time` is set to *01:00*, `prov.polling.timeRandomEnd` is set to *05:00*, and you have 25 phones, a random subset of those 25 phones, as determined by the MAC address, will poll randomly between 1am and 5am every day.
- If `prov.polling.mode` is set to `abs` and `prov.polling.period` is set to *2328000*, the phone polls every 20 days.

Provisional Polling of the CX5500 System

Central Provisioning Server	**template** > parameter
To enable polling and set the mode, period, time, and time end parameters	**site.cfg** > prov.polling.*

Example Provisional Polling Configuration

The following illustration shows the default sample random mode configuration for the provisional polling feature in the **site.cfg** template file. In this setup, every phone will poll once per day, between 1 and 5 am.



Tip: Only provision files when polling

If prov.startupCheck.enabled="0" then the CX5500 system will not look for the sip.ld or the configuration files when they are rebooted, lose power, or restarted. Instead, they will look only when receiving a checksync message, a polling trigger, or a manually started update from the menu or web UI.
Some files such as bitmaps, .wav, the local directory and any custom ringtones will still be downloaded every time as they are stored in RAM and lost with every reboot.

Set Up Microsoft Lync Server 2010 and 2013

Microsoft® Lync® Server 2010 and 2013 each provide a unified communications (UC) solution that enables customers, colleagues, and business partners to communicate instantly by voice, video, or messaging through a single interface, regardless of their location or network. The following features are available with the CX5500 system registered with Lync Server.

- **Shared Line Appearance** Assign administrative delegates to answer, hold, and transfer calls, set distinct ringtones, and make calls on behalf of Boss lines.
- **Lync Management** Sign in and out of Lync using your login credentials or PIN authentication, set your presence status, manage your Lync contacts, and search for contacts in the Lync directory.
- **Address Book Service (ABS)** Access and search a complete corporate directory.
- **Call Park:** Call park enables you to place a call on a separate line, called a call orbit, where anyone can retrieve the call.

Polycom CX5500 software enables you to register a single phone line with Lync Server; you cannot register multiple or shared lines with Lync Server.

The section following, Registering with Microsoft Lync Server 2010, provides an important overview of Polycom provisioning methods and an example configuration to get a phone registered with Lync Sever.

For details on the user features available on Polycom phones registered with Microsoft Lync Server 2010, see *Feature Profile 72430: Using Polycom® Phones with Microsoft® Lync™ Server 2010*.

For details on the user features available on Polycom phones registered with Microsoft Lync Server 2010, see *Feature Profile 84538: Using Polycom® VVX® Business Media Phones with Microsoft® Lync™ Server 2013*.

Note: You must purchase a license to use Microsoft Lync Server 2010 with the CX5500 System.

You must purchase a *Lync Feature License* from a Polycom reseller or Polycom sales representative to use Polycom products in a Microsoft Lync environment. You can use the CX5500 system in a Lync environment for trial purposes, without purchasing a license, to a maximum of 30 days.

The concurrent failover/fallback feature explained in Set Up Server Redundancy is not compatible with Microsoft Lync Server.

Note: Understanding the Lync Contact List and Your Phone's Local Contact Directory

When you are running CX5500 software for use with Lync Server 2010, you have access to two separate contact lists: the default local contact directory on your CX5500 system and a Lync contact list. If you want to disable the local contact directory on your CX5500 system or make it read-only, see Use the Local Contact Directory.

Register with Microsoft Lync Server 2010

You can register the CX5500 system with Lync Server 2010 in one of three ways:

- Using the Web Configuration Utility
- Using centralized provisioning, which includes a provisioning server and configuration files in XML format.
- From the phone user interface

Note: Registering a Phone with Lync Server 2010

For details on using the phone user interface and for details on each registration method, including registration instructions, see *Deploying Polycom® UC Software for use with Microsoft® Lync™ Server 2010*.

Set the Base Profile to Lync - Phone User Interface and Web Configuration Utility

You can quickly register phones with the Lync Server by setting the phone's Base Profile to *Lync* from the phone's user interface or using the Web Configuration Utility. Note that although registering the phone using either of these two methods is simpler than centralized provisioning, each method registers one phone at a time. In addition, you cannot enable extensive diagnostic logging that the phone writes to the provisioning server, contact directory files, or phone user interface language files.

Centralized Provisioning

You can register multiple phones to Lync Server using a provisioning server and configuration files in XML format. You can provision your phones with Lync Server 2010 using the **lync.cfg** template configuration file included with Polycom CX5500 softwre. Polycom recommends using this method - also called centralized provisioning - when deploying multiple phones, about twenty or more. A provisioning server enables you to store configuration files in a single location on a server, which simplifies maintenance of feature settings and updates for multiple phones. In addition, use of a provisioning server allows the phones to send diagnostic and other information to files stored on the server, including log files, a contact directory, individual call lists, and multiple languages on the phone user interface.

Ensure Security

The CX5500 systems are computing devices that you must configure for security as you do other computing devices. Polycom strongly recommends that you change the default user name and password on each Polycom device on first deployment. To maximize security, do not leave user name and password fields blank. Create user names and passwords of a reasonably long length, and change user names and passwords periodically.

Polycom provides the following ways for you to change the administrative password of a device:

- Configuration File
- Web Configuration Utility
- Phone User Interface
- CX5100/CX5500 Control Panel

Configuration File

Polycom provides configuration files in XML format that you can use to change user names and passwords. You can modify the attached sample configuration file and add it to your file directory, or you can add the parameters and values directly to your existing configuration files. However you use the files or parameters, ensure that you add them to your boot server directory. After you have updated you configuration files, you need to update your device configuration from the device user interface by going to **Settings > Basic > Update Configuration**.

Settings: Use a Secure Protocol

Use a secure provisioning protocol such as FTPS or HTTPS to maximize security of user names and passwords.

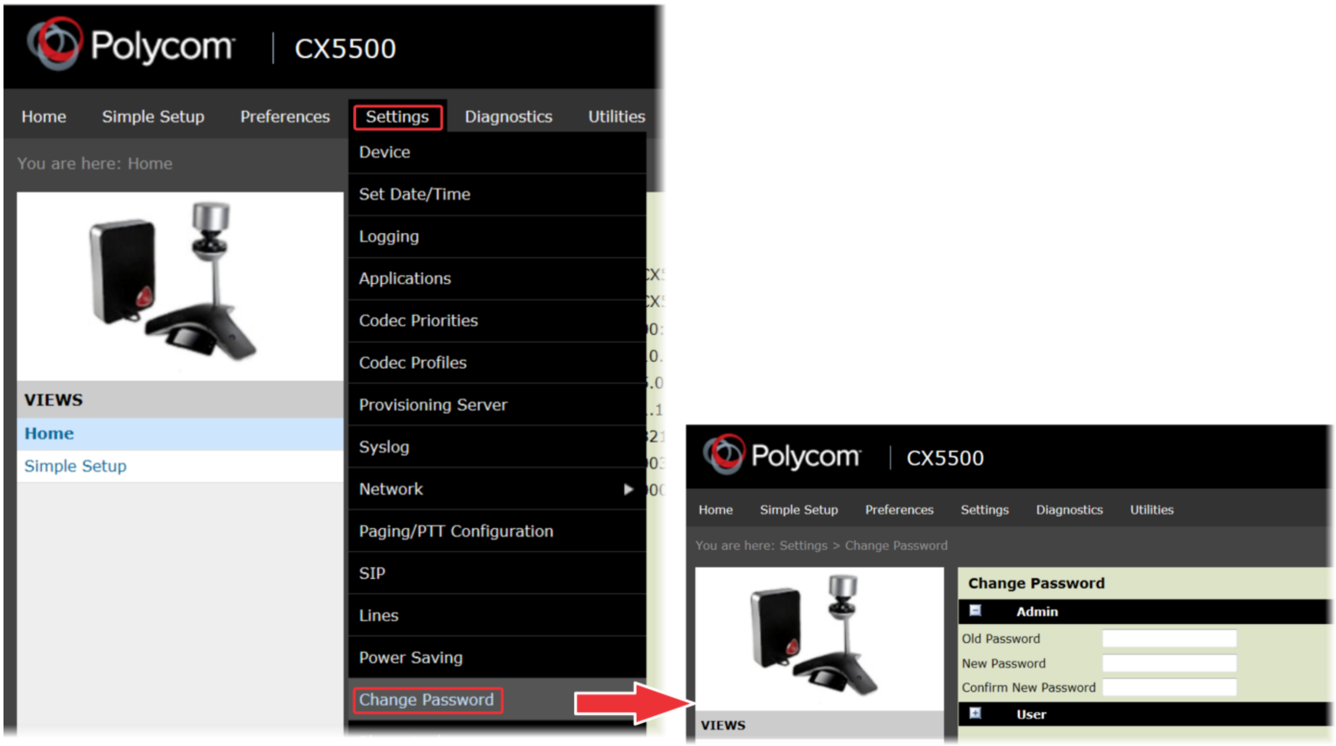
Web Configuration Utility

The Web Configuration Utility enables you to configure settings and features on a per-phone basis. To access the Web Configuration, enter the IP address of the device to the address bar of your browser. Log in as Admin and enter the default password 456.

Settings: Use HTTPS

Polycom recommends using the Web Configuration Utility with HTTPS to maximize security.

In the Web Utility, go to **Settings** > **Change Password** to access settings that change the user name and password, as shown next.



Phone User Interface

On your phone, select **Settings > Advanced**, enter the default password **456**, and tap **Administration Settings > Change Admin Password**.

Example Configuration: Setting the Base Profile to Lync

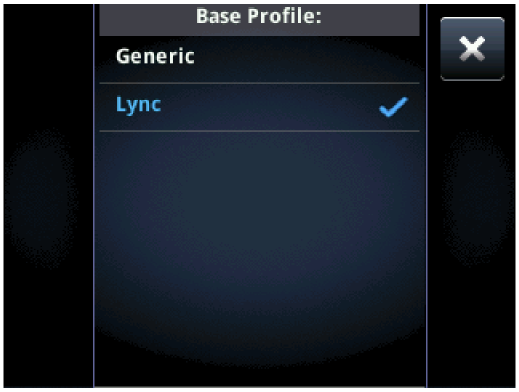
This example configuration shows you how to set the phone's Base Profile to *Lync* using the phone's interface. For instructions on all methods you can use to provision CX5500 systems with Lync Server, including tips on how to quickly provision multiple phones to save time, see the Polycom Lync Provisioning Guide.

When you set the phone Base Profile to Lync you are provisioning the phone with the minimum number of parameters required to register a CX5500 system with Lync Sever 2010. However, if your organization's security procedures don't allow you to enter user IDs and password in cleartext to configuration files set `reg.x.auth.useLoginCredentials` to `1` and instruct each user to enter their credentials through the phone's user interface—the Login Credential screen.

To set the Base Profile to Lync:

1. Tap **Settings > Advanced**.
2. Enter the password (default 456) and press **Enter**.
3. Tap **Administration Settings > Network Configuration** and scroll to **Base Profile.**

4 In the **Base Profile** menu, select **Lync**, as shown.



The phone automatically restarts and displays the Lync Server *Sign In* screen.

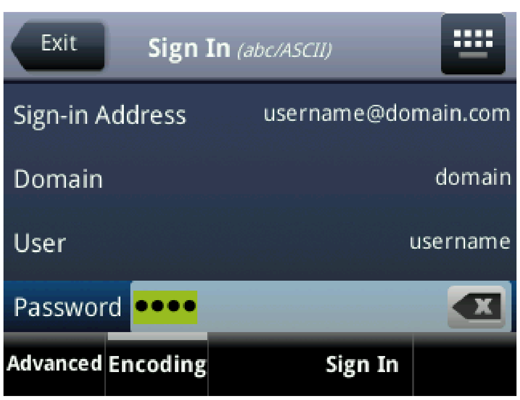
Troubleshooting: Rebooting the Phone

If the phone does not restart, you can manually restart by powering off/on the phone. You can also manually reboot the phone: Tap **Settings > Advanced**, enter the password (default **456**), and choose **Reboot Phone**. When the phone completes the reboot cycle, the Lync Server Sign In screen displays.

To sign in and register a line with Lync Server:

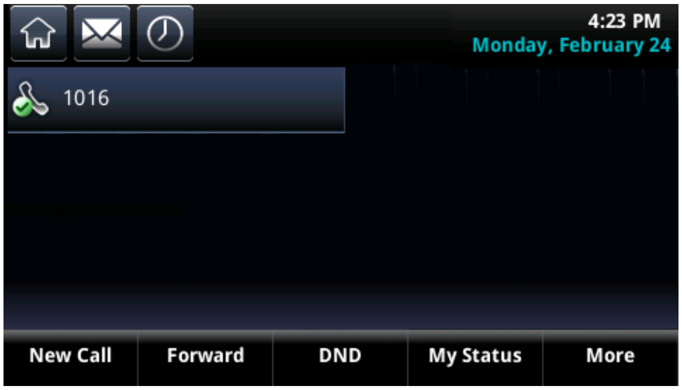
1 Enter your sign in credentials in the following formats:

- **Sign In Address** This is your Lync SIP URI address, not the user name for the Active Directory account. For example, *username@domain.com*.
- **Domain** By default, use the NetBIOS *domain* name. If that does not work, try the DNS domain name (for example, *domain.com)*.
- **User** user name
- **Password** *password*



2 Select **Sign In**.

The phone registers with Lync Server and you can begin using Lync features directly from the phone. The following illustration shows a line 1, extension *1016* on the CX5500 system successfully registered to Lync Server.



There are two ways to sign in/out of Lync:

- Tap **Settings > Features > Microsoft Lync > Sign In/Sign Out**.
- Press the **More** soft key and select the **Sign In/Sign Out** soft key.

Admin Tip: Workaround for Phones using G.722 and Retrieving Microsoft Lync Voicemail

If your CX5500 systems are configured with G.722 and users find that they do not hear audio when retrieving voicemail from the Microsoft Lync Server, you need to make the following changes to parameters in the site.cfg template file:
Change `voice.codecPref.G7221.24kbps` from 0 to 5.
Change `voice.codecPref.G7221.32kbps` from 5 to 0.
Add `voice.audioProfile.G7221.24kbps.payloadType` and set it to 112.

Enable Microsoft Exchange Calendar Integration

The CX5500 system can display the Microsoft Exchange 2007 and 2010 calendar. The calendar gives you quick access to meeting information and you can dial in to conference calls. To integrate the Microsoft Exchange Calendar features with your phone, configure the parameters in the table Enable Microsoft Exchange Calendar Integration.

You can launch the feature from a calendar icon that displays in Home view or in the Features menu.

You need a valid Microsoft Windows credentials to access the Microsoft Exchange Calendar information on the phone. You can manage these credentials through the Login Credentials, which are available through **Settings > Basic > Login Credentials**.

You can view the calendar information in day or month format. The meeting details also display beside the calendar view.

All possible phone numbers that you can dial to place a call to the meeting display in the meeting details. You can automatically place a call by pressing a soft key.

A reminder pop-up is displayed 15 minutes before a scheduled meeting. You can dismiss the reminder, select snooze to have the reminder pop up again, open the meeting details view. A tone will be played along with the reminder pop-up.

Web Info: Using Microsoft Exchange Calendar Integration

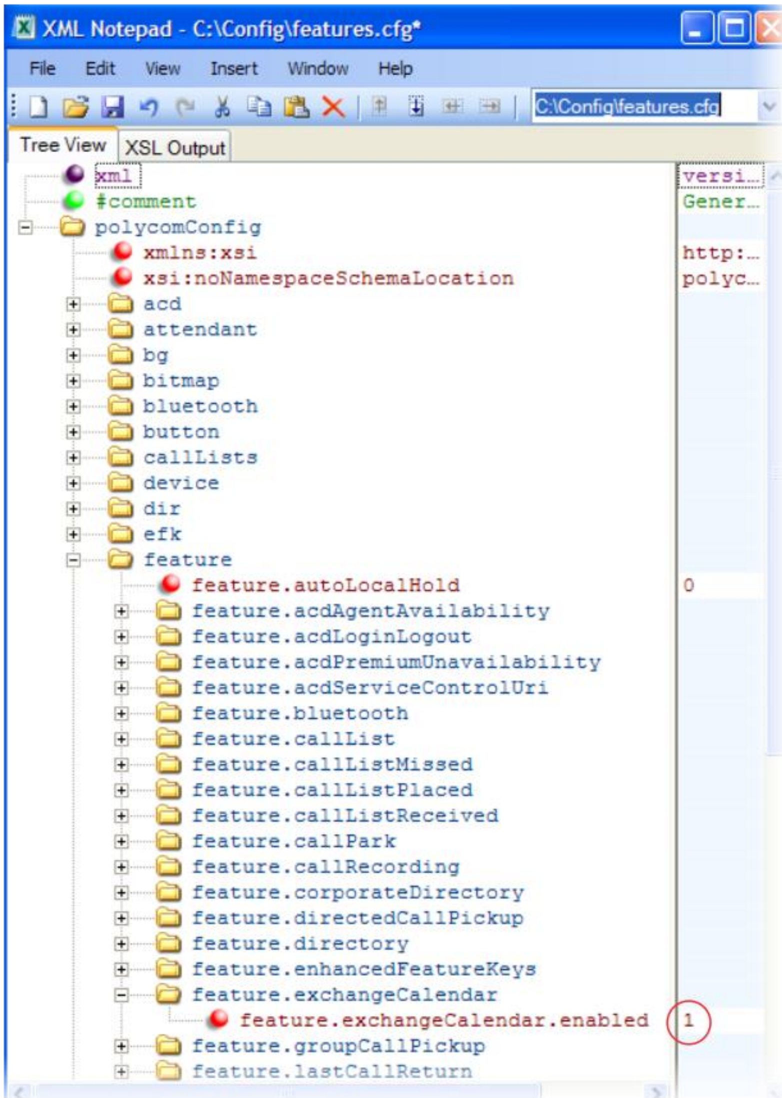
For user instructions on how to use calendar integration, refer to the *Polycom CX5500 Unified Conference Station User Guide*.

Enable Microsoft Exchange Calendar Integration

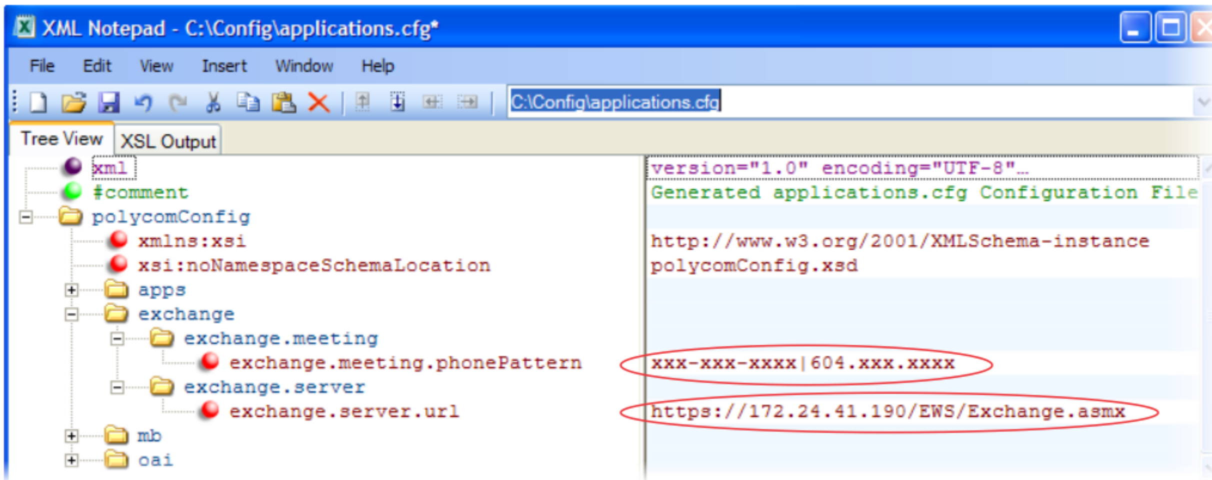
Central Provisioning Server	template > parameter
Turn Microsoft Exchange Calendar Integration on or off	**features.cfg** > feature.exchangeCalendar.enabled
Specify the Microsoft Exchange Server address	**applications.cfg** > exchange.server.url
Specify the pattern to use to identify phone numbers in meeting descriptions	**applications.cfg** > exchange.meeting.phonePattern
Turn the meeting reminder on or off	**applications.cfg** > exchange.meeting.reminderEnabled
Web Configuration Utility	
To enable Microsoft Exchange Calendar Integration and configure the settings, navigate to **Settings > Applications** and expand **Exchange Applications**.	

Example Exchange Calendar Configuration

The following example shows the Calendar feature enabled in **features.cfg**.



After you enable the feature, specify the Microsoft Exchange Server address in **applications.cfg**, as shown next. In this example, a pattern has been specified for meeting numbers. When you specify a pattern, any number in your meeting invitation that matches the pattern will display on a meeting participants' phones as a soft key. Then, participants can press the soft key to dial in to the meeting. You can specify multiple patterns, separated by a bar. In the following example, two patterns are specified.



Set Up Phone Audio Features

After you set up your Polycom® phones on the network, phone users can send and receive calls using the default configuration. However, you might consider modifications that optimize the audio quality of your network.

Frequency bandwidth is one of the most critical elements affecting the intelligibility of speech in telephony. The frequency range that the human ear is most sensitive to is far beyond the capabilities of the plain old telephony system (POTS). In fact 80 percent of the frequencies in which speech occurs are not even used by public telephone networks because they only operate from 300Hz to 3.5 kHz. Complicating the intelligibility of telephony speech in today's world is background noise, variations in environmental reverberation, and communication among persons speaking a variety of native languages. While VoIP technology can broaden the frequency bandwidth and improve sound quality and intelligibility, it can also increase the network load and create a demand for lower raw bit rates. As Audio Codec Specifications shows, Polycom offers phones with a range of codecs, including codecs with high frequency bandwidth and low raw bit rates.

This section describes the audio sound quality features and options you can configure for your CX5500 system. Use these features and options to optimize the conditions of your organization's phone network system.

This section shows you how to update your configuration for the following audio-related features:

- Customize Audio Sound Effects   Enables you to customize sound effects associated with incoming calls and other events.
- Voice Activity Detection   Conserves network bandwidth by detecting periods of relative 'silence' in the transmit data path and replacing that silence with special packets that indicate silence is occurring.
- Generate Dual Tone Multi-Frequency (DTMF) Tones   Generates dual tone multi-frequency (DTMF) tones in response to user dialing on the dial pad.
- DTMF Event RTP Payload   Conforms to RFC 2833, which describes a standard RTP-compatible technique for conveying DTMF dialing and other telephony events over an RTP media stream.
- Acoustic Echo Cancellation   Employs advanced acoustic echo cancellation for handsfree operation.
- Audio Codecs   Enables access to a wide range of industry standard audio codecs.
- IP Type-of-Service   Enables the setting packet priority.
- IEEE 802.1p/Q   The phone may tag all Ethernet packets it transmits with an 802.1Q VLAN header.
- Voice Quality Monitoring (VQMon)   Generates various quality metrics including MOS and R-factor for listening and conversational quality. This feature is part of the Productivity Suite

This section also outlines the following built-in audio processing features, which do not require any configuration changes to work:

- Automatic Gain Control   Designed for handsfree operation, this feature boosts the transmit gain of the local user in certain circumstances.
- Background Noise Suppression   Designed primarily for handsfree operation, this feature reduces background noise to enhance communication in noisy environments.

- Comfort Noise Fill   Provides a consistent noise level to the remote user of a handsfree call.
- Dynamic Noise Reduction   Provides maximum microphone sensitivity, while automatically reducing background noise. The CX5500 system automatically supports this non-adjustable feature. This feature is also known as Noise Suppression.
- Jitter Buffer and Packet Error Concealment   Employs a high-performance jitter buffer and packet error concealment system designed to mitigate packet inter-arrival jitter, and out-of-order, lost, or delayed packets.
- Low-Delay Audio Packet Transmission   Minimizes latency for audio packet transmission.

To troubleshoot any problems with your CX5500 system on the network, see Miscellaneous Maintenance Tasks. For more information on the configuration files, see Configuration Methods. For more information on the Web Configuration Utility, see Provision with the Web Configuration Utility. For instructions on how to read the feature descriptions in this section, see Read the Feature Parameter Tables.

Customize Audio Sound Effects

You can customize the audio sound effects that are used for incoming calls and other alerts using synthesized tones or sampled audio files. You can replace the default sampled audio files with your own custom **.wav** audio file format. The phone supports the following **.wav** audio file formats:

- mono G.711 (13-bit dynamic range, 8-khz sample rate)
- mono L16/16000 (16-bit dynamic range, 16-kHz sample rate)
- mono L16/32000 (16-bit dynamic range, 32-kHz sample rate)
- mono L16/44100 (16-bit dynamic range, 44.1 kHz sample rate)
- mono L16/48000 (16-bit dynamic range, 48-kHz sample rate)

Note: Supported Audio Formats

The L16/32000 and L16/48000 wav formats are supported only on the CX5500 system.

Your custom sampled audio files must be available at the path or URL specified by `saf.x` in the table Customize Audio Sound Effects so the phone can download them. Include the name of the file and the **.wav** extension in the path.

Customize Audio Sound Effects

Central Provisioning Server	template > parameter
Specify a path or URL for the phone to download a custom audio file	**site.cfg** > saf.x
Specify the name, type, and value for a custom sound effect	**region.cfg** > se.pat.*
Web Configuration Utility	
To add, play, or delete a custom audio file, navigate to **Settings > Basic > Preferences > Ringtones** and expand the **Custom Audio Files** menu.	

Example Configuration

The following example configuration illustrates how to add a custom sound effect from a sampled audio file. In the example, the custom audio files MyTone.wav and Chirp.wav have been added as sound effects 12 and 13. The `welcome` sound has been customized to use the sampled audio file 13 (*Chirp.wav*) with the label *Birds*. Ringtone 19 is named *Whistle* and is configured to use sampled audio file *12* (*MyTone.wav*).

Voice Activity Detection

The purpose of voice activity detection is to detect periods of silence in the transmit data path so the phone doesn't have to transmit unnecessary data packets for outgoing audio. This process conserves network bandwidth. The VAD parameters in the table Voice Activity Detection (VAD) will help you set up this feature. For compression algorithms without an inherent VAD function, such as G.711, the phone uses the codec-independent comfort noise transmission processing specified in RFC 3389. The RFC 3389 algorithm is derived from G.711 Appendix II, which defines a comfort noise (CN) payload format (or bit-stream) for G.711 use in packet-based, multimedia communication systems. The phone generates CN packets—also known as Silence Insertion Descriptor (SID) frames—and also decodes CN packets, to efficiently regenerate a facsimile of the background noise at the remote end.

Voice Activity Detection (VAD)

Central Provisioning Server	template > parameter
Specify if G.729 Annex B should be signaled	**site.cfg** > voice.vad.signalAnnexB
Enable or disable voice activity detection	**site.cfg** > voice.vadEnable

Specify the threshold between active voices and background voices	**site.cfg** > voice.vadThresh

Generate Dual Tone Multi-Frequency (DTMF) Tones

The phone generates dual tone multi-frequency (DTMF) tones in response to user dialing on the dial pad. Use the parameters in the table Dual Tone Multi-Frequency (DTMF) Tone Generation to set up this feature. These tones, commonly referred to as *touch tones,* are transmitted in the real-time transport protocol (RTP) streams of connected calls. The phone can encode the DTMF tones using the active voice codec or using RFC 2833-compatible encoding. The coding format decision is based on the capabilities of the remote endpoint.

Dual Tone Multi-Frequency (DTMF) Tone Generation

Central Provisioning Server	**template** > parameter
Specify if DTMF tones should be played through the speakerphone	**sip-interop.cfg** > tone.dtmf.chassis.masking
Specify the frequency level of DTMF digits	**sip-interop.cfg** > tone.dtmf.level
Specify how long the phone should wait between DTMF digits	**sip-interop.cfg** > tone.dtmf.onTime
Specify how long the phone should play each DTMF tone for	**sip-interop.cfg** > tone.dtmf.onTime
Enable or disable DTMF encoding in an RTP stream	**sip-interop.cfg** > tone.dtmf.viaRtp

DTMF Event RTP Payload

The phone is compatible with *RFC 2833—RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals*. RFC 2833 describes a standard RTP-compatible technique for conveying DTMF dialing and other telephony events over an RTP media stream. The phone generates RFC 2833 (DTMF only) events but does not regenerate—or otherwise use—DTMF events received from the remote end of the call. Use the parameters in the table DTMF Event RTP Payload to set up this feature.

DTMF Event RTP Payload

Central Provisioning Server	**template** > parameter
Specify if the phone will use RFC 2833 to encode DTMF	**sip-interop.cfg** > tone.dtmf.rfc2833Control
Specify the phone-event payload encoding in the dynamic range to be used in SDP offers	**sip-interop.cfg** > tone.dtmf.rfc2833Payload

Acoustic Echo Cancellation

Your CX5500 system uses advanced acoustic echo cancellation (AEC). See the table Audio Codecs Supported on the CX5500 System for a list of audio codecs available for the CX5500 system and their priority. The system uses both linear and non-linear techniques to aggressively reduce echo while permitting natural, full-duplex communication patterns.

Caution: Contact Polycom Support Before Modifying Acoustic Echo Cancellation Parameters

Consult Polycom Support before you make changes to any acoustic echo cancellation parameters.

Audio Codecs

The following table lists the audio codecs supported on the CX500 system.

Audio Codecs Supported on the CX5500 System

Codec	*Priority*
G.722.1C.48kbps	2
G.722.1C.32kbps	0
G.722.1C.24kbps	0
Siren14.48kbps	3
Siren14.32kbps	0
Siren14.24kbps	0
G.722.1.32kbps	5
G.722.1.24kbps	0
G.722.1.16kbps	0
G.719.64kbps	0
G.719.48kbps	0
G.719.32kbps	0
G.722	4
G.711Mu	6
G.711A	7

Codec	*Priority*
G.729AB	8
Lin16.48ksps	0
Lin16.44.1ksps	0
Lin16.32ksps	0
Lin16.16ksps	0
Lin16.8ksps	0

The table Audio Codec Specifications summarizes the audio codecs supported on the CX5500 system:

Audio Codec Specifications

Algorithm	*Reference*	*Raw Bit Rate*	*IP Bit Rate*	*Sample Rate*	*Default Payload Size*	*Effective Audio Bandwidth*
G.719	RFC 5404	32 Kbps 48 Kbps 64 Kbps	48 Kbps 64 Kbps 80 Kbps	48 Ksps	20 ms	20 KHz
G.711	RFC 1890	64 Kbps	80 Kbps	16 Ksps	20 ms	7 KHz
G.722.1	RFC 3047	16 Kbps 24 Kbps 32 Kbps	32 Kbps 40 Kbps 48 Kbps	16 Ksps	20 ms	7 KHz
G.722.1C	G7221C	224 Kbps 32 Kbps 48 Kbps	40 Kbps 48 Kbps 64 Kbps	32 Ksps	20 ms	14 KHz
G.729AB	RFC 1890	8 Kbps	24 Kbps	8 Ksps	20 ms	3.5 KHz
Lin16	RFC 1890	128 Kbps 256 Kbps 512 Kbps 705.6 Kbps 768 Kbps	132 Kbps 260 Kbps 516 Kbps 709.6 Kbps 772 Kbps	8 Ksps 16 Ksps 32 Ksps 44.1 Ksps 48 Ksps	10 ms	3.5 KHz 7 KHz 14 KHz 20 KHz 22 KHz
Siren14	SIREN14	24 Kbps 32 Kbps 48 Kbps	40 Kbps 48 Kbps 64 Kbps	32 Ksps	20 ms	14 KHz

Note: Network Bandwidth Requirements for Encoded Voice

The network bandwidth necessary to send the encoded voice is typically 5–10% higher than the encoded bit rate due to packetization overhead. For example, a G.722.1C call at 48 kbps for both the receive and transmit signals consumes about 100 kbps of network bandwidth (two-way audio).

Use parameters in the table Audio Codec Priorities to specify the priority for audio codecs.

Audio Codec Priorities

Central Provisioning Server	**template** > parameter
To specify the priority for a codec	**site.cfg** > voice.codecPref.<nameOfCodec>
Web Configuration Utility	
To enable or disable codecs and specify codec priority, navigate to **Settings > Codec Profiles** and expand the **Audio Priority** menu.	

IP Type-of-Service

The *type-of-service* field in an IP packet header consists of four type-of-service (TOS) bits and a 3-bit precedence field. See the table IP Type-of-Service (ToS) for available parameters. Each TOS bit can be set to either 0 or 1. The precedence field can be set to a value from 0 through 7. The type of service can be configured specifically for RTP packets and call control packets, such as SIP signaling packets.

IP Type-of-Service (ToS)

Central Provisioning Server	**template** > parameter
Set the IP header bits for call control	**site.cfg** > qos.ip.callControl.*
Set the IP header bits for RTP	**site.cfg** > qos.ip.rtp.*
Set the IP header bits for RTP video	**site.cfg** > qos.ip.rtp.video.*
Web Configuration Utility	
Set the QoS IP settings by navigating to **Settings > Network > QoS**.	

IEEE 802.1p/Q

The phone will tag all Ethernet packets it transmits with an 802.1Q VLAN header when:

- A valid VLAN ID specified in the phone's network configuration.
- The phone is instructed to tag packets through Cisco Discovery Protocol (CDP) running on a connected Ethernet switch.
- A VLAN ID is obtained from DHCP or LLDP (see DHCP Menu).

Use the table IEEE 802.1p/Q to set values. The 802.1p/Q `user_priority` field can be set to a value from 0 to 7. The `user_priority` can be configured specifically for RTP packets and call control packets, such as SIP signaling packets, with default settings configurable for all other packets.

IEEE 802.1p/Q

Central Provisioning Server	template > parameter
Set the user priority for packets without a per-packet protocol setting (including 802.1p/Q)	**site.cfg** > qos.ethernet.other.user_priority
Web Configuration Utility	
To set the user priority for 802.1p/Q packets, navigate to **Settings > Network > QoS** and expand the **Other Protocols** menu.	

Voice Quality Monitoring (VQMon)

You can configure the phones to generate various quality metrics you can use to monitor sound and listening quality. These metrics can be sent between the phones in RTCP XR packets, which are compliant with RFC 3611—RTP Control Extended Reports (RTCP XR). The packets are sent to a report collector as specified in draft RFC draft-ietf_sipping_rtcp-summary-02. The metrics can also be sent as SIP PUBLISH messages to a central voice quality report collector.

A license key is required to activate the VQMon feature on all phones. For more information on VQMon, contact your Certified Polycom Reseller.

You can enable three types of voice quality reports:

- **Alert** Generated when the call quality degrades below a configurable threshold.
- **Periodic** Generated during a call at a configurable period.
- **Session** Generated at the end of a call.

You can generate a wide range of performance metrics, the parameters for which are shown in Voice Quality Monitoring (VQM). Some are based on current values, such as jitter buffer nominal delay and round trip delay, while others cover the time period from the beginning of the call until the report is sent, such as network packet loss. Some metrics are computed using other metrics as input, such as listening Mean Opinion Score (MOS), conversational MOS, listening R-factor, and conversational R-factor.

Voice Quality Monitoring (VQM)

Central Provisioning Server	template > parameter
Specify the warning threshold for alerts	**features.cfg** > voice.qualityMonitoring.collector.alert.*
Enable the generation of quality reports	**features.cfg** > voice.qualityMonitoring.collector.enable.*
Specify the server address and port	**features.cfg** > voice.qualityMonitoring.collector.server.x.*
Enable the generation of RTCP-XR packets	**features.cfg** > voice.qualityMonitoring.rtcpxr.enable

Built-In Audio Processing Features

Your CX5500 system has the following built-in audio processing features: automatic gain control, background noise suppression, comfort noise fill, dynamic noise reduction, jitter buffer and packet error concealment, and low delay audio packet transmission. These features work automatically, without configuration changes.

Automatic Gain Control

Automatic Gain Control (AGC) is applicable to handsfree operation and is used to boost the transmit gain of the local talker in certain circumstances. This increases the effective user-phone radius and helps with the intelligibility of soft-talkers.

Background Noise Suppression

Background noise suppression (BNS) is designed primarily for handsfree operation and reduces background noise to enhance communication in noisy environments.

Comfort Noise Fill

Comfort noise fill is designed to help provide a consistent noise level to the remote user of a handsfree call. Fluctuations in perceived background noise levels are an undesirable side effect of the non-linear component of most AEC systems. This feature uses noise synthesis techniques to smooth out the noise level in the direction toward the remote user, providing a more natural call experience.

Dynamic Noise Reduction

Dynamic noise reduction (DNR) provides maximum microphone sensitivity, while automatically reducing background noise— from fans, projectors, heating and air conditioning—for clearer sound and more efficient conferencing.

Jitter Buffer and Packet Error Concealment

The phone employs a high-performance jitter buffer and packet error concealment system designed to mitigate packet inter-arrival jitter and out-of-order, or lost or delayed (by the network) packets. The jitter buffer is adaptive and configurable for different network environments. When packets are lost, a concealment algorithm minimizes the resulting negative audio consequences.

Low-Delay Audio Packet Transmission

The phone is designed to minimize latency for audio packet transmission.

Set Up User and Phone Security Features

After you set up your CX5500 system on your network with the default configuration, users can place and answer calls. Polycom's Open SIP UC software enables you to make custom configurations to optimize security settings.

This section shows you how to update your configuration for the following security features:

- Local User and Administrator Passwords Several local settings menus are protected with two privilege levels—user and administrator—each with its own password.
- Incoming Signaling Validation Levels of security are provided for validating incoming network signaling.
- Configuration File Encryption Confidential information stored in configuration files can be protected (encrypted). The phone can recognize encrypted files, which it downloads from the provisioning server, and it can encrypt files before uploading them to the provisioning server.
- Digital Certificates The CX5500 system supports digital certificates and associated private keys.
- Generate a Certificate Signing Request Create a request to obtain a device certificate.
- TLS Profiles Configure your phone with a profile that specifies trusted digital certificates. You can also install and specify custom certificates.
- Support Mutual TLS Authentication Support phone authentication of the server and server authentication of the phone.
- Configurable TLS Cipher Suites Control which of cipher suites will be offered/accepted during TLS session negotiation.
- Secure Real-Time Transport Protocol Encrypting audio streams to avoid interception and eavesdropping. Encrypting audio streams to avoid interception and eavesdropping.
- Lock the Phone Prevent access to the phone menu and to key presses.
- Support 802.1X Authentication Authenticate devices connecting to a local area network (LAN) or a wireless local area network (WLAN).
- Set User Profiles Access your personal phone settings from any phone in your organization's network.

To troubleshoot any problems with your CX5500 system on the network, see Troubleshoot Your CX5500 System.

For more information on the configuration files, see Use the Centralized Provisioning Method - Configuration Files.

For more information on the Web Configuration Utility, see Provision with the Web Configuration Utility.

For instructions on how to read the parameter tables for features listed in this section, see Read the Feature Parameter Tables.

Local User and Administrator Passwords

Several local settings menus are protected with user and administrator passwords. The phone will prompt you for a user or administrator password before you can access certain menu options. If the phone requires the administrator password, you may be able to use the user password, but you will be presented with limited menu options. If the phone prompts you for the user password, you may use the administrator password (you will see the same menus as the user). The Web Configuration Utility is protected by the user and administrator password and displays different features and options depending on which password you use. The default user password is **123** and the default administrator password is **456**. You should change the administrator password from the default value. You may want to change the user password for security reasons, see the table Local User and Admininstrator Password Settings for all parameters.

Local User and Administrator Password Settings

Central Provisioning Server	**template** > parameter
Set the minimum length for the administrator password	**site.cfg** > sec.pwd.length.admin
Set the minimum length for the user password	**site.cfg** > sec.pwd.length.user
Set the phone's local administrator password	**device.cfg** > device.auth.localAdminPassword
Set the phone's local user password	**device.cfg** > device.auth.localUserPassword
Web Configuration Utility	
To change the user or administrator password, navigate to **Settings > Change Password**. To change the administrator password, you must log in to the Web configuration utility as an administrator.	
Local Phone User Interface	
To change the administrator password, navigate to **Settings > Advanced**, enter the current administrator password, and select **Admin Settings** > **Change Admin Password**. To change the User Password, navigate to **Settings > Advanced**, enter the current user or administrator password, and select **Change User Password**.	

Incoming Signaling Validation

You can choose from three optional levels of security for validating incoming network signaling:

- Source IP address validation
- Digest authentication
- Source IP address validation and digest authentication

See the table Incoming Signaling Validation Parameters for the parameters that specify the validation type, method, and the events you want to validate.

Incoming Signal Validation Parameters

Central Provisioning Server	template > parameter
Specify what type of validation to perform	**sip-interop.cfg** > voIp.SIP.requestValidation.x.method
Set the name of the method for which validation will be applied	**sip-interop.cfg** > voIp.SIP.requestValidation.x.request
Determine which events within the Event header should be validated	**sip-interop.cfg** > voIp.SIP.requestValidation.x.request.y.event

Configuration File Encryption

You can encrypt configuration files, contact directories, and configuration override files can all be encrypted. Note that you cannot encrypt the master configuration file.

You can determine whether encrypted files are the same as unencrypted files and use the SDK to facilitate key generation. Use the table Configuration File Encryption Parameters to configure the parameters used to encrypt files. For more information about encrypting configuration files, see Encrypting Configuration Files.

Configuration File Encryption Parameters

Central Provisioning Server	template > parameter
Specify if configuration files uploaded from the phone to the provisioning server should be encrypted	**site.cfg** > sec.encryption.upload.config
Specify if the contact directory is encrypted when it is uploaded from the phone to the provisioning server	**site.cfg** > sec.encryption.upload.dir
Specify if the configuration overrides file should be encrypted when it is uploaded from the phone to the server	**site.cfg** > sec.encryption.upload.overrides
Specify an encryption key so the phone can download encrypted files from the provisioning server	**device.cfg** > device.sec.configEncryption.key

‘

Digital Certificates

You can download the Polycom Root CA from http://pki.polycom.com/. The certificate is set to expire on March 9, 2044.

Web Info: Digital Certificates on Polycom Phones

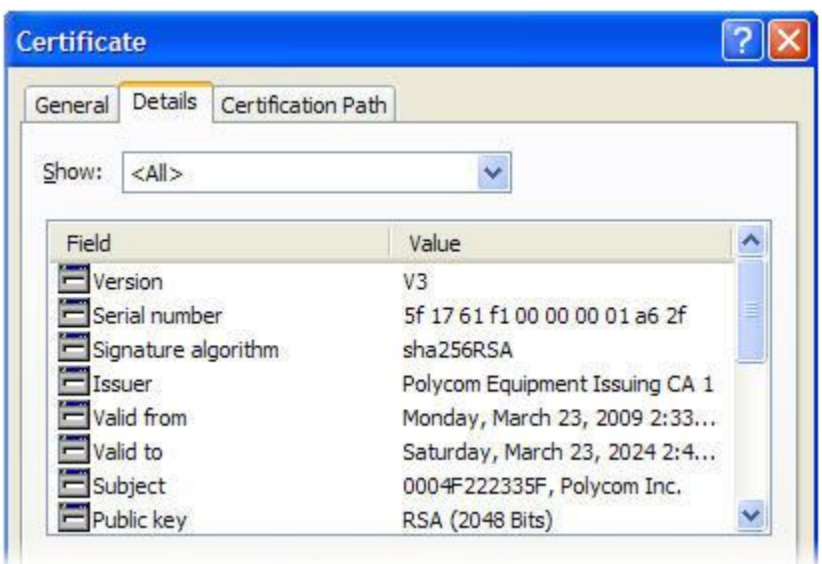
For details on installing digital credentials on all phones, see *Feature Profile 37148: Device Certificates on Polycom SoundPoint IP, SoundStation IP, and VVX Phones*.

Polycom uses the X.509 standard, which defines what information can go into a certificate. An X.509 digital certificate is a digitally signed statement. All X.509 certificates have the following fields, in addition to the signature:

- **Version** This identifies which version of the X.509 standard applies to this certificate, which in turn affects what information can be specified in the certificate.
- **Serial Number** The entity that created the certificate is responsible for assigning it a serial number to distinguish it from other certificates it issues.
- **Signature Algorithm Identifier** This identifies the algorithm used by the Certificate Authority (CA) to sign the certificate.
- **Issuer Name** The X.500 name of the entity that signed the certificate. This is normally a CA. Using this certificate means trusting the entity that signed this certificate.
- **Validity Period** Each certificate is valid for a limited amount of time. This period is described by a start date and time and an end date and time, and can be as short as a few seconds or almost as long as a century.
- **Subject Name** The name of the entity whose public key the certificate identifies. This name uses the X.500 standard, so it is intended to be unique across the Internet.
- **Subject Public Key Information** This is the public key of the entity being named, together with an algorithm identifier that specifies to which public key cryptographic system this key belongs and any associated key parameters.

Polycom supports the use of Subject Alternative Names (SAN) with TLS security certificates. Polycom does not support the use of the asterisk (*) or wildcard characters in the Common Name field of a Certificate Authority’s public certificate. If you want to enter multiple hostnames or IP addresses on the same certificate, use the SAN field.

The following is an example of a Polycom device certificate when viewed in a browser.



The device certificate and associated private key are stored on the phone in its non-volatile memory as part of the manufacturing process. For more information on digital certificates, see Public Key Infrastructure (X.509) and RFC 2459: Internet X.509 Public Key Infrastructure .

Web Info: Using Custom Device Certificates With Polycom Phones

As of UC Software 4.0.0, you can install custom device certificates on your Polycom phones. These certificates are installed in the same way custom CA certificates are installed. See *Technical Bulletin 17877: Using Custom Certificates With Polycom Phones*.

To determine if there is a device certificate on a CX5500 system:

1. Tap **Settings > Advanced > Admin Settings > TLS Security > Custom Device Certificates**.

   You can view the Polycom device certificate on the phone at **Settings > Status > Platform** > **Phone**.

2. Tap the **Info** soft key to view the certificate.

   One of the following messages will be displayed:

   - **Device Certificate: Installed** or **Device Certificate: Factory Installed** is displayed if the certificate is available in flash memory, all the certificate fields are valid (listed above), and the certificate has not expired.
   - **Device Certificate: Not Installed** is displayed if the certificate is not available in flash memory (or the flash memory location where the device certificate is to be stored is blank).
   - **Device Certificate: Invalid** is displayed if the certificate is not valid.

Note: Device Certificate Shown as Self-Signed

Some Polycom phones manufactured after December, 2011 report the device certificate as 'self-signed' and not as 'Factory Installed'. The difference indicates that different issuing CAs were used to generate the certificates. As long as the authenticating server trusts the Polycom Root CA that issued these certificates, the phones will operate correctly.

Generating a Certificate Signing Request

You may need a certificate to perform a number of tasks, for example, multiple TLS authentication. To obtain a certificate you need to:

- Request a certificate from a Certificate Authority (CA) by creating a certificate signing request (CSR).
- Forward the CSR to a CA to create a certificate. If your organization doesn't have its own CA, you will need to forward the CSR to a company like Symantec. If successful, the CA will send back a certificate that has been digitally signed with their private key.

After you receive the certificate, you can download it to the phone:

- Using a configuration file
- Through the phone's user interface
- Through the Web Configurable Utility

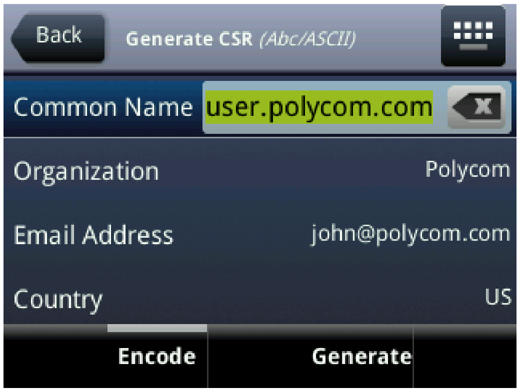
To generate a certificate signing request on a CX5500 system:

1 Navigate to **Settings > Advanced > Admin Settings > Generate CSR**.

1 When prompted, enter the administrative password and press the **Enter** soft key. The default administrative password is **456**.

2 From the **Generate CSR Screen**, fill in the **Common Name** field - the Organization, Email Address, Country, and State fields are optional.

The following figure shows the Generate CSR screen.



3 Press **Generate**.

A message *CSR generation completed* displays on the phone's screen. The MAC.csr (certificate request) and MAC-private.pem (private key) are uploaded to the phone's provisioning server.

Configure TLS Profiles

The Transport Layer Security (TLS) profiles describe a collection of custom CA and device certificates installed on the CX5500 systems and the features where these certificates are used for authentication.

Your phone can trust certificates issued by widely recognized certificate authorities when trying to establish a connection to a provisioning server for application provisioning. There are a number of parameters you can use to configure TLS Profiles listed in TLS Platform Profile and TLS Application Profile Parameters For the complete list of trusted Certificate Authorities, see Trusted Certificate Authority List.

Custom CA and device certificates can be added to the phone and set up to be used by different features. For example, the phone's factory-installed or custom device certificate could be used for authentication when phone provisioning is performed by an HTTPS server. A custom CA certificate could also be used when accessing content through the microbrowser or browser.

Once you install certificates on the phone, you can to determine which TLS Platform Profiles or TLS Application Profiles will use these certificates. By default, TLS Platform Profile 1 uses every CA certificate and the default device certificate. Also, each TLS Application uses TLS Platform Profile 1 as the default profile. You can quickly apply a CA certificate to all TLS Applications by installing it on the phone and keeping the default TLS Profile and default TLS Application values.

Lastly you must choose which TLS platform profile or application profile will be used for each TLS Application. The profiles can be used for phone provisioning, with the applications running on the microbrowser and browser, and for 802.1X, LDAP, and SIP authentication. Some applications, such as Syslog, can only use a TLS Platform Profile, not a TLS Application Profile. See <TLS/> for the list of applications.

For more information on device (or digital) certificates installed on the phones at the factory, see Digital Certificates.

Web Info: Using Custom CA Certificates

For more information on using custom certificates, see *Technical Bulletin 17877: Using Custom Certificates With Polycom Phones*.

The following table shows parameters for TLS Platform Profile 1. To configure TLS Platform Profile 2, use a *2* at the end of the parameter instead of a *1*. For example, set `device.sec.TLS.profile.caCertList2` instead of `.caCertList1`.

TLS Platform Profile and TLS Application Profile Parameters

Central Provisioning Server	**template** > parameter
TLS Platform Profile Parameters (use 2 at the end of each parameter (instead of 1) to set up platform profile 2)	
Specify which CA certificates to use	**device.cfg** > device.sec.TLS.profile.caCertList1
Specify the cipher suite	**device.cfg** > device.sec.TLS.profile.cipherSuite1
Select the default cipher suite or a custom cipher suite	**device.cfg** > device.sec.TLS.profile.cipherSuiteDefault1
Specify a custom certificate	**device.cfg** > device.sec.TLS.customCaCert1
Specify which device certificates to use	**device.cfg** > device.sec.TLS.profile.deviceCert1
TLS Application Profile Parameters	
Specify which CA certificates to use	**site.cfg** >sec.TLS.profile.x.caCert.*
Specify the cipher suite	**site.cfg** >sec.TLS.profile.x.cipherSuite
Select the default cipher suite or a custom cipher suite	**site.cfg** >sec.TLS.profile.x.cipherSuiteDefault
Specify a custom certificate	**site.cfg** > sec.TLS.customCaCert.x
Specify which device certificates to use	**site.cfg** > sec.TLS.profile.x.deviceCert
Specify the custom device key	**site.cfg** > sec.TLS.customDeviceKey.x

Web Configuration Utility

To install CA or device certificates and configure TLS profiles, navigate to **Settings > Network > TLS** and expand the **Certificate Configuration** and **TLS Profiles** menus.

Local Phone User Interface

To install a CA or device certificate, navigate to **Settings > Advanced > Admin Settings > TLS Security** and select **Custom CA Certificates** or **Custom Device Credentials** and enter the URL of a custom certificate or PEM-encoded certificate.
Once you have configured the certificates, configure a TLS profile. To configure TLS profiles, navigate to **Settings > Advanced > Admin Settings > TLS Security > Configure TLS Profiles**. Select the profile that you would like to configure, and configure the cipher suite, choose which CA certificates to use, and choose which device certificates to use. The menu options are: Configure Cipher Suite, CA Certificates, and Device Certificates.

This section provides detailed information on:

- Download Certificates to a CX5500 System
- Set TLS Profiles

Download Certificates to a CX5500 System

You can download certificates to a CX5500 system by specifying a URL where the certificate is currently stored. You can install up to eight CA certificates and eight device certificates on the phone. You can refresh certificates when they expire or are revoked. You can delete any CA certificate or device certificate that you install.

Note: Maximum Size for Certificates

The maximum certificate size on Platform CA1 is 1536KB and 4KB for Platform CA2.

To download a certificate to a CX5500 system:

1 Navigate to **Settings > Advanced > Administrative Settings > TLS Security** and select **Custom CA Certificates** or **Custom Device Certificates**.

When prompted, enter the administrative password and tap the **Enter** soft key. The default administrative password is **456**.

2 Select the **Install** soft key.

3 Enter the URL where the certificate is stored.

For example, *http://bootserver1.vancouver.polycom.com/ca.crt*

4 Select the **Enter** soft key.

The certificate is downloaded. The certificate's MD5 fingerprint displays to verify that the correct certificate is to be installed.

5 Select the **Accept** soft key.

The certificate is installed successfully.

The appropriate certificate menu displays the certificate's common name.

Set TLS Profiles

By default, all Polycom-installed profiles are associated with the default cipher suite and use trusted and widely recognized CA certificates for authentication. Use the table Set a TLS Profile for each TLS Application to set parameters. You can change the cipher suite, CA certificates, and device certificates for the two platform profiles and the six application profiles. You can then map profiles directly to the features that use certificates.

Set a TLS Profile for each TLS Application

Central Provisioning Server	**template** > parameter
Specify the TLS profile to use for each application (802.1X and Provisioning)	**device.cfg** > device.sec.TLS.profileSelection.*
Specify the TLS profile to use for each application (other applications)	**device.cfg** >sec.TLS.profileSelection.*
Web Configuration Utility	
To specify the TLS profile to use for a specific application, navigate to **Settings > Network > TLS**, and expand the **TLS Applications menu**.	
Local Phone User Interface	
To specify the TLS profile to use for a specific application, navigate to **Settings > Advanced > Admin Settings > TLS Security > TLS Applications**, select the **TLS application**, and choose a **TLS Profile** to use.	

Support Mutual TLS Authentication

Mutual Transport Layer Security (TLS) authentication is a process in which both entities in a communications link authenticate each other. In a network environment, the phone authenticates the server and vice-versa. In this way, phone users can be assured that they are doing business exclusively with legitimate entities and servers can be certain that all would-be users are attempting to gain access for legitimate purposes.

This feature requires that the phone being used has a Polycom factory-installed device certificate or a custom device certificate installed on it. See the section, Digital Certificates.

Prior to SIP 3.2, and in cases where the phones do not have device certificates, the phone will authenticate to the server as part of the TLS authentication, but the server cannot cryptographically authenticate the phone. This is sometimes referred to as Server Authentication or single-sided Authentication.

Mutual TLS authentication is optional and is initiated by the server. When the phone acts as a TLS client and the server is configured to require mutual TLS, the server will request and then validate the client certificate during the handshake. If the server is configured to require mutual TLS, a device certificate and an associated private key must be loaded on the phone.

The device certificate, stored on the phone, is used by:

- HTTPS device configuration, if the server is configured for Mutual Authentication

- SIP signaling, when the selected transport protocol is TLS and the server is configured for Mutual Authentication
- Syslog, when the selected transport protocol is TLS and the server is configured for Mutual Authentication
- Corporate Directory, when the selected transport protocol is TLS and the server is configured for Mutual Authentication
- 802.1X Authentication, if the server is configured for Mutual Authentication (optional for EAP-TLS)

Note: You Cannot Modify the Factory-Installed Certificate or Private Key

Users cannot modify or update the digital certificate or the associated private key installed on the phone during manufacturing. Users can install a custom device certificate to be used instead of, or in addition to, the factory-installed certificate.

The Polycom Root CA can be downloaded from http://pki.polycom.com. The location of the Certificate Revocation List (CRL)—a list of all expired certificates signed by the Polycom Root CA—is part of the Polycom Root CA digital certificate. If Mutual TLS is enabled, the Polycom Root CA or your organization's CA must be downloaded onto the HTTPS server.

The following operating system/Web server combinations have been tested and verified:

- Microsoft Internet Information Services 6.0 on Microsoft Windows Server 2003
- Apache v1.3 on Microsoft Windows XP

Web Info: Provisioning Using Microsoft Internet Information Services

For more information on using Mutual TLS with Microsoft® Internet Information Services (IIS) 6.0, see *Engineering Advisory 52609: Mutual Transport Layer Security Provisioning Using Microsoft Internet Information Services 6.0*.

Configurable TLS Cipher Suites

The phone administrator can control which cipher suites will be offered/accepted during TLS session negotiation. The phone supports the cipher suites listed in the table TLS Cipher Suites and you can use the parameers listed in Configurable TLS Cipher Suites to configure TLS Cipher Suites. The 'Null Cipher' listed in the following table is a special case option which will not encrypt the signaling traffic, and is useful for troubleshooting purposes.

TLS Cipher Suites

Cipher	*Cipher Suite*
ADH	ADH-RC4-MD5, ADH-DES-CBC-SHA, ADH-DES-CBC3-SHA, ADH-AES128-SHA, ADH-AES256-SHA
AES128	AES128-SHA

Cipher	*Cipher Suite*
AES256	AES256-SHA
DES	DES-CBC-SHA, DES-CBC3-SHA
DHE	DHE-DSS-AES128-SHA, DHE-DSS-AES256-SHA, DHE-RSA-AES128-SHA, DHE-RSA-AES256-SHA
EXP	EXP-RC4-MD5, EXP-DES-CBC-SH, EXP-EDH-DSS-DES-CBC-SHA, EXP-DES-CBC-SHA, EXP-ADH-RC4-MD5, EXP-ADH-DES-CBC-SHA, EXP-EDH-RSA-DES-CBC-SHA
EDH	EDH-RSA-DES-CBC-SHA, EDH-DSS-DES-CBC3-SHA, EDH-DSS-CBC-SHA
NULL	NULL-MD5, NULL-SHA
RC4	RC4-MD5, RC4-SHA

Tip: Changes to the Default TLS Cipher Suites in UC Software 4.0.0

Changes have been made to the default TLS cipher suites in UC Software 4.0.0. If you created customized TLS cipher suites in a previous release of the UC Software, your changes will be lost unless you backup the configuration files.

Configurable TLS Cipher Suites

Central Provisioning Server	**template** > parameter
Specify the global cipher list	**site.cfg** > sec.TLS.cipherList
Specify the cipher list for a specific TLS Platform Profile or TLS Application Profile	**site.cfg** > sec.TLS.<application>.cipherList
Web Configuration Utility	
To specify the cipher list for a specific TLS Platform Profile or TLS Application Profile, navigate to **Settings > Network > TLS** and expand the **TLS Profiles menu**.	
Local Phone User Interface	
To specify the cipher list for a specific TLS Platform Profile or TLS Application Profile, navigate to **Settings > Advanced > Admin Settings > TLS Profiles > Configure TLS Profiles**, select a profile, and choose **Configure Cipher Suite**.	

Secure Real-Time Transport Protocol

Secure Real-Time Transport Protocol (SRTP) provides a way of encrypting audio stream(s) to avoid interception and eavesdropping on phone calls. As described in RFC 3711, both RTP and RTCP signaling may be encrypted using an AES (advanced encryption standard) algorithm. The parameters used to configure SRTP are shown in Secure Real Time Transport Protocol Parameters. When this feature is enabled, phones negotiate with the other end-point the type of encryption and authentication to use for the session. This negotiation process is compliant with RFC4568—Session Description Protocol (SDP) Security Descriptions for Media Streams.

Web Info: SRTP RFC Resources

For more information on SRTP, see RFC 3711. For the procedure describing how two phones set up SRTP for a call, see RFC 4568.

Authentication proves to the phone receiving the RTP/RTCP stream that the packets are from the expected source and have not been tampered with. Encryption modifies the data in the RTP/RTCP streams so that, if the data is captured or intercepted, it sounds like noise and cannot be understood. Only the receiver knows the key to restore the data.

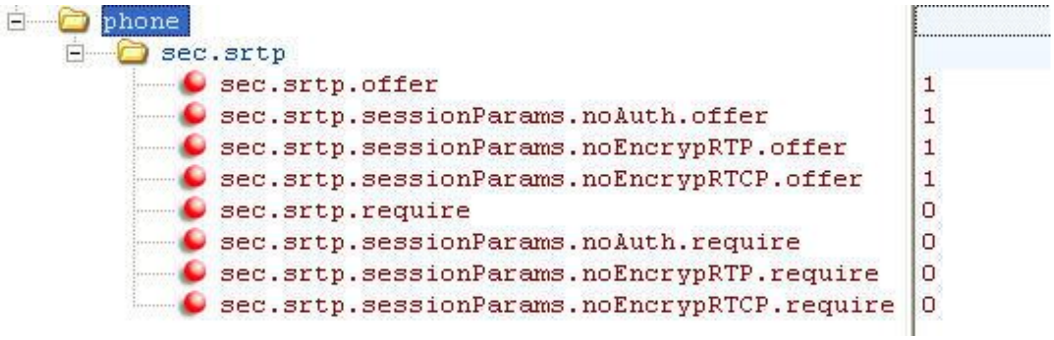
A number of session parameters have been added to enable you to turn off authentication and encryption for RTP and RTCP streams. This is done mainly to reduce the phone's processor usage.

If the call is completely secure (RTP authentication and encryption and RTCP authentication and RTCP encryption are enabled), then the user sees a padlock symbol appearing in the last frame of the connected context animation (two arrows moving towards each other)

Secure Real Time Transport Protocol Parameters

Central Provisioning Server	**template** > parameter
Enable SRTP	**sip-interop.cfg** > sec.srtp.enable
Include secure media in SDP of SIP INVITE	**sip-interop.cfg** > sec.srtp.offer
Include crypto in offered SDP	**sip-interop.cfg** > sec.srtp.offer.*
Secure media stream required in all SIP INVITEs	**sip-interop.cfg** > sec.srtp.require
Check tag in crypto parameter in SDP	**sip-interop.cfg** > sec.srtp.requireMatchingTag
Specify if the phone offers and/or requires: RTP encryption, RTP authentication, and RTCP encryption	**sip-interop.cfg** > sec.srtp.sessionParams.*

In the following example, the **srtp_1.cfg** configuration file is shown below:



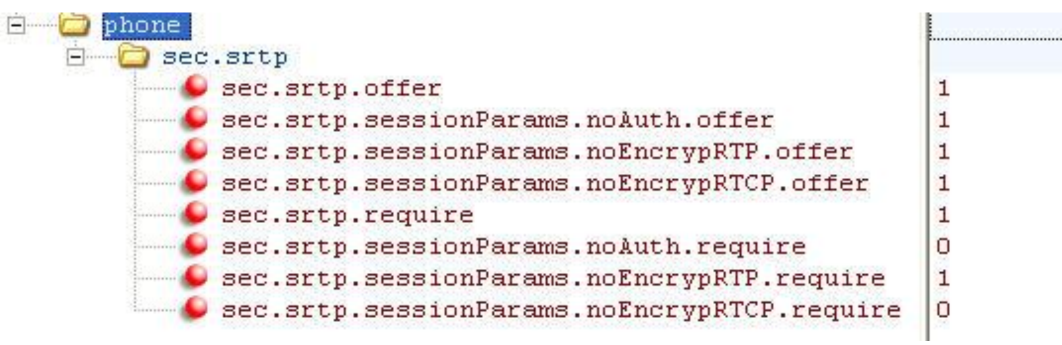
This would result in an offer (SIP INVITE with SDP) with 8 crypto attributes with the following session parameters:

`<no session parameters> UNENCRYPTED_SRTCP UNENCRYPTED_SRTP`
`UNAUTHENTICATED_SRTP`

`UNAUTHENTICATED_SRTP,UNENCRYPTED_SRTCP UNENCRYPTED_SRTP,UNENCRYPTED_SRTCP`
`UNAUTHENTICATED_SRTP,UNENCRYPTED_SRTP`
`UNAUTHENTICATED_SRTP,UNENCRYPTED_SRTP,UNENCRYPTED_SRTCP`

In the above example, the crypto attributes are ordered "most secure" to "least secure" (more security turned off). The phone receiving this call should chose the most secure crypto it can support based on the SRTP *require* settings in **sip.cfg** and reply with it in the SDP of a 200 OK SIP message.

In this example, the **srtp_2.cfg** configuration file is shown below:



This results in an offer (SIP INVITE with SDP) with 4 crypto attributes with the following session parameters:

`UNENCRYPTED_SRTP UNENCRYPTED_SRTP,UNENCRYPTED_SRTCP`
`UNAUTHENTICATED_SRTP,UNENCRYPTED_SRTP`
`UNAUTHENTICATED_SRTP,UNENCRYPTED_SRTP,UNENCRYPTED_SRTCP`

In the above example, every crypto includes the `UNENCRYPTED_SRTP` session parameter because it is required.

If nothing compatible is offered based on the receiving phone's STRP "require" settings, then the call is rejected or dropped.

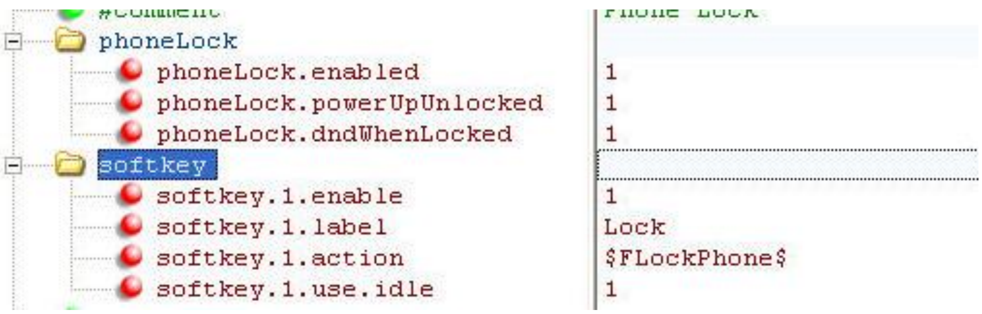
Lock the Phone

As of Polycom UC Software 3.3.0, users can lock their phones, and prevent access to the menu or key presses, by tapping the **Lock** soft key or through the phone menu.

Note: Displaying the Lock Soft Key On Your Phone

You need to enable the enhanced feature key (EFK) feature if you want your phone to display a Lock soft key. See feature.enhancedFeatureKeys.enabled.

The following configuration file snippet shows how to display the **Lock** soft key.



Once the phone is locked, all user features and access to menus are disabled. The messages “The phone is locked.” and “Authorized calls only.” display on the screen. Incoming calls to the phone may receive a Do Not Disturb message. You can specify the authorized numbers to which users can place calls.

Using the **New Call** soft key, users can place calls using up to five authorized numbers including the emergency number. If the user places a call —using the keypad— to a number that matches an authorized number, the call will proceed. This is to ensure that certain numbers such as emergency numbers can be placed from the phone.

To unlock the phone, the user presses the **Unlock** soft key and enters their password; if it is entered correctly, the phone returns to its normal idle state.

In case the user forgets their password, the system administrator can unlock their phone either by entering the administrator password or by disabling (and re-enabling) the phone lock feature. The latter method facilitates remote unlocking and avoids disclosing the administrator password to the user. See the table Phone Lock Parameters for the parameters that configure the phone lock feature.

Note: Shared Lines on Locked Phones

If a locked phone has a registered shared line, calls to the shared line will be displayed on the locked phone and the phone’s user can answer the call.

Phone Lock Parameters

Central Provisioning Server	template > parameter
Enable enhanced feature keys	**features.cfg** > feature.enhancedFeatureKeys.enabled
Enable or disable phone lock	**features.cfg** > phoneLock.enabled
Specify an authorized contact (description and value) who can be called while the phone is locked	**features.cfg** > phoneLock.authorized.*
Specify the scenarios when phone lock should be enabled	**features.cfg** > phoneLock.*
Web Configuration Utility	
To enable and configure phone lock, navigate to **Settings > Phone Lock**.	
Local Phone User Interface	
To lock the phone, press the Lock soft key (if available) or navigate to **Settings > Basic > Preferences > Lock Phone**. To unlock the phone, press the **Unlock** soft key and enter the user or administrator password.	

Support 802.1X Authentication

IEEE 802.1X is a port-based Network Access Control (PNAC). It provides an authentication mechanism to devices trying to attach to a local area network (LAN) or a wireless local area network (WLAN). IEEE 802.1X is based on the Extensible Authentication Protocol (EAP). The figure A Typical 802.1X Network Configuration shows a typical 802.1X network configuration with wired and wireless CX5500 systems.

A Typical 802.1X Network Configuration

The CX5500 system supports the following EAP authentication methods:

- EAP-TLS (requires Device and CA certificates)
- EAP-PEAPv0/MSCHAPv2 (requires CA certificates)
- EAP-PEAPv0/GTC (requires CA certificates)
- EAP-TTLS/MSCHAPv2 (requires CA certificates)
- EAP-TTLS/GTC (requires CA certificates)
- EAP-FAST (optional Protected Access Credential (PAC) file, if not using in-band provisioning)
- EAP-MD5

To set up an EAP method that requires a Device or CA certificate, you need to configure TLS Platform Profile 1 or TLS Platform Profile 2 to use with 802.1X. You can use the parameters in the table Set 802.1X Authentication Parameters to configure 802.1X Authentication. For more information see TLS Profiles.

Web Info: EAP Authentication Protocol

For more information, see RFC 3748: Extensible Authentication Protocol.

Set 802.1X Authentication Parameters

Central Provisioning Server	**template** > parameter
Enable or disable the 802.1X feature	**device.cfg** > device.net.dot1x.enabled
Specify the identity (username) for authentication	**device.cfg** > device.net.dot1x.identity
Specify the 802.1X EAP method	**device.cfg** > device.net.dot1x.method
Specify the password for authentication	**device.cfg** > device.net.dot1x.password
To enable EAP In-Band Provisioning for EAP-FAST	**device.cfg** > device.net.dot1x.eapFastInBandProv
Specify a PAC file for EAP-FAST (optional)	**device.cfg** > device.pacfile.data
Specify the optional password for the EAP-FAST PAC file	**device.cfg** > device.pacfile.password
Web Configuration Utility	
To enable and configure the 802.1X feature, navigate to **Settings > Network > Ethernet** and expand the **Ethernet 802.1X** menu.	
Local Phone User Interface	
To enable 802.1X authentication, navigate to the Ethernet Menu (**Settings > Advanced > Admin Settings > Network Configuration > Ethernet Menu**) and select **802.1X Auth**. To configure the 802.1X feature, navigate to the **Ethernet Menu** and select **802.1X Menu** (802.1X Auth must be set to enable first).	

Set User Profiles

There are a number of parameters shown in the table User Profile Parameters that enable users to access their personal phone settings from any phone in the organization. This means that users can access their contact directory and speed dials, as well as other phone settings, even if they temporarily change work areas. This feature is particularly useful for remote and mobile workers who do not have a dedicated work space and conduct their business in more than one location. The User Profile feature is also beneficial if an office has a common conference phone. In this case, multiple users could use the phone and access their own settings.

If a user changes any settings while logged in to a phone, the settings will be saved and displayed the next time the user logs in to a phone. When a user logs out, the user's personal phone settings are no longer displayed.

If you set up the User Profile feature, a user can log in to a phone by entering their user ID and password. The default password is **123**.

Tip: Calling Authorized Numbers while Logged Out

You can configure the phones so that anyone can call authorized and emergency numbers when not logged in to a phone. For more information, see dialplan.routing.emergency.outboundIdentity.

If the User Profile feature is set up on your company's phones, users can:

- Log in to a phone to access their personal phone settings.
- Log out of a phone after they finish using it.
- Place a call to an authorized number from a phone that is in the logged out state.
- Change their user password.

When you set up the User Profile feature, you will have to decide whether you want to require users to always log in to a phone. If the User Profile feature is enabled, but not required, users can choose to use the phone as is (that is, without access to their personal settings), or they can log in to display their personal settings. You can specify if a user is logged out of the phone when the phone restarts or reboots, or if they remain logged in.

You can also choose to define default credentials for the phone (see the section Create a Phone Configuration File). If you specify a default user ID and password, the phone automatically logs itself in each time an actual user logs out or the phone restarts or reboots. When the phone logs itself in using the default login credentials, a default phone profile is displayed (as defined in the phone's master configuration file on the provisioning server). In this scenario, users will still have the option to log in and view their personal settings.

To set up the User Profile feature, perform the following procedures on the provisioning server:

- Create a phone configuration file, or update an existing file, to enable the feature's settings.
- Create a user configuration file—called ***<user>*.cfg**—that specifies the user's password and registration, and other user-specific settings that you want to define.

Tip: Resetting a User's Password

You can reset a user's password by removing the password parameter from the override file. This will cause the phone to use the default password in the <*user*>.cfg file.

After you complete these procedures, update the phone's configuration to affect your changes. The User Profile feature will be ready to use.

User Profile Parameters

Central Provisioning Server	**template** > parameter
Enable or disable the user profile feature	**site.cfg** > prov.login.enabled
Specify the amount of time before a non-default user is logged out	**site.cfg** > prov.login.automaticLogout
Specify the default password for the default user	**site.cfg** > prov.login.defaultPassword
Specify if the phone can have users other than the default user	**site.cfg** > prov.login.defaultOnly
Specify the name of the default user	**site.cfg** > prov.login.defaultUser
Specify the password used to validate the user login	**site.cfg** > prov.login.localPassword
Specify if a user should remain logged in after the handset reboots	**site.cfg** > prov.login.persistent
Specify if a user must log in while the feature is enabled	**site.cfg** > prov.login.required

Create a Phone Configuration File

Create a phone configuration file for the User Login feature, and then add and set the attributes for the feature. Or, if you already have a phone configuration file, update the file to include the User Login parameters you want to change.

Tip: Creating a Default User Password for All Users

Polycom recommends that you create a single default user password for all users.

To define the feature's settings:

1. Create a **site.cfg** file for the phone and place it on the provisioning server.

   You can base this file on the sample configuration template that is in your software package. To find the file, navigate to ***<provisioning server location>*/Config/site.cfg**.

2. In **site.cfg**, open the `<prov.login/>` attribute, and then add and set values for the user login attributes.

The following example is an example **site.cfg** file. Your file will contain different values, depending on how you want the feature to work.

prov.login	
prov.login.automaticLogout	0
prov.login.defaultDomain	
prov.login.defaultOnly	0
prov.login.defaultPassword	
prov.login.defaultUser	
prov.login.enabled	0
prov.login.localPassword	123
prov.login.persistent	0
prov.login.required	0

Create a User Configuration File

Create a configuration file for each user that you want to be able to log in to the phone. The name of the file will specify the user’s login ID. In the file, specify any user-specific settings that you want to define for the user.

Tip: Converting a Phone-Based Deployment to a User-Based Deployment

To convert a phone-based deployment to a user-based deployment, copy the <*MACaddress*>-phone.cfg file to <*user*>-phone.cfg and copy phoneConfig<*MACaddress*>.cfg to <*user*>.cfg.

To create a user configuration file:

1 On the provisioning server, create a user configuration file for each user that will be able to log in to the phone. The name of the file will be the user’s ID to log in to the phone. For example, if the user’s login ID is **user100**, the name of the user’s configuration file is **user100.cfg**.

2 In each **<user>.cfg** file, you can add and set values for the user’s login password (optional).

3 Add and set values for any user-specific parameters, such as:

- Registration details (for example, the number of lines the profile will display and line labels).
- Feature settings (for example, microbrowser settings).

Caution: Adding User-Specific Parameters

If you add optional user-specific parameters to <user>.cfg, add only those parameters that will not cause the phone to restart or reboot when the parameter is updated. For information on which parameters cause the phone to restart or reboot, see the Configuration Parameters.

The following is a sample user configuration file.

If a user updates their password or other user-specific settings using the Main Menu on the phone, the updates will be stored in **<*user*>-phone.cfg**, not **<*MACaddress*>-phone.cfg**.

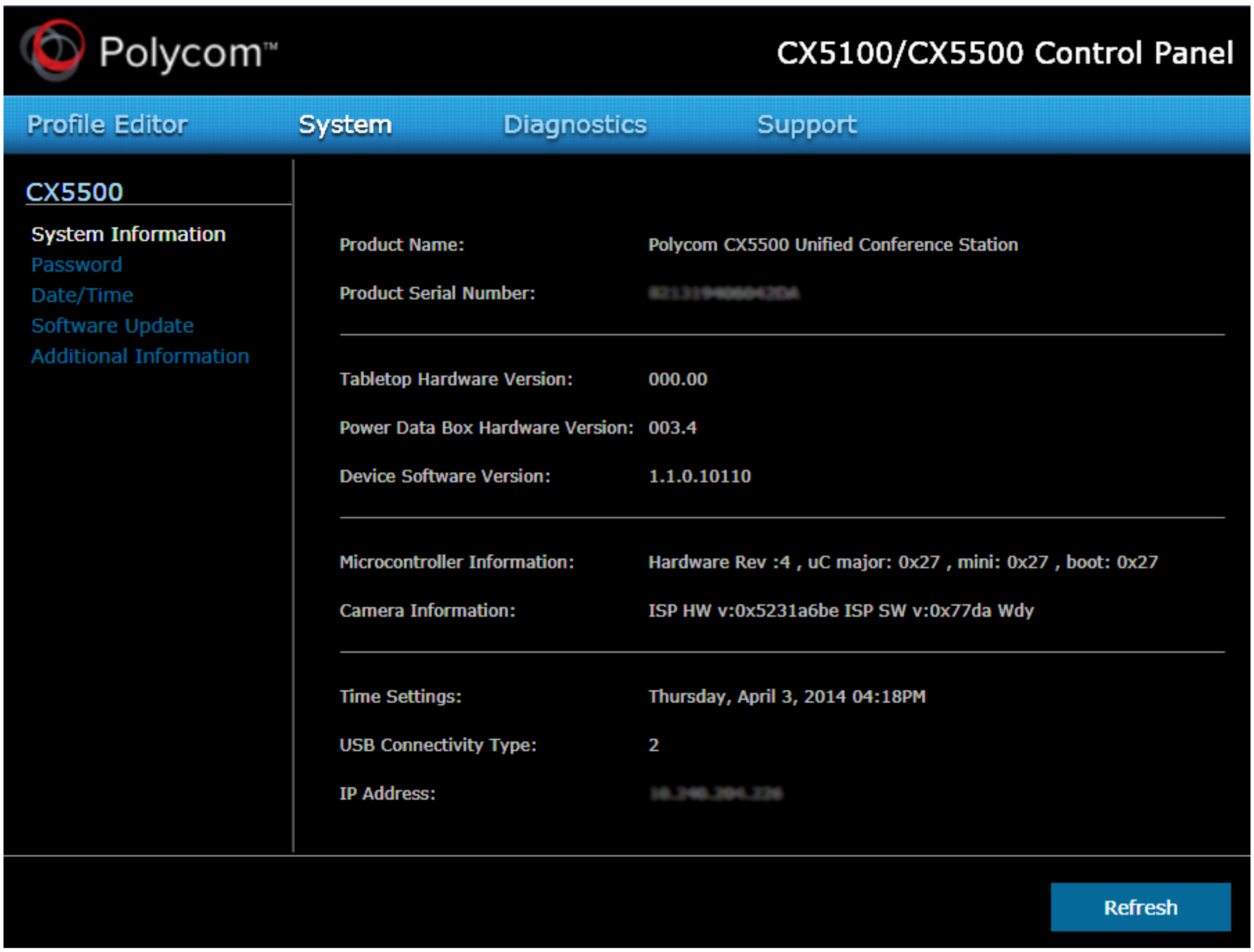
If a user updates their Contact Directory while logged in to a phone, the updates will be stored in **<user>-directory.xml**. Directory updates will be displayed each time the user logs in to a phone. For certain phones, an up-to-date call lists history will be defined in **<user>-calls.xml**. This list will be retained each time the user logs in to their phone. Configuration parameter precedence (from first to last) for a phone that has the User Profile feature enabled is:

- **<user>-phone.cfg**
- Web Configuration Utility (through a browser)
- Polycom CMA system
- Configuration files listed in the master configuration file (including **<*user*>.cfg**)
- Default values

Use the CX5100/5500 Control Panel

The Polycom CX5100/CX5500 Control Panel enables you to change a limited group of settings for an individual system when connected to a computer and used as a video conference device. If you are not using the telephony features of the CX5500 system, you can use the Control Panel to configure your system. Note that you cannot configure telephony settings and features in the Control Panel.

You can download and install the Control Panel from the Polycom Support site. The following figure shows the System Information tab in the Control Panel.



The Control Panel provides a user-friendly, intuitive method to configure settings for using the CX5500 as a connected device.

After you install the Control Panel, you can connect your system to your computer and create a profile for CX5500 system, view your system's information, change system settings, and view diagnostics and retrieve logs.

Find Your Default System Password

To make changes to your Polycom CX5500 system using the Control Panel, enter the system password. By default, the password is the 14-digit system serial number. You can find the serial number on the label on the back panel of the power data box, as shown in the following figure.

Location of the Serial Number Label on the Power Data Box

After you enter the default password, you can change the system's password in the System tab in the Control Panel.

To change the system default password:

1 In the Control Panel, click **System > Password**.

2 Enter the default password in the **Old Password** field.

3 Enter a new password for the system in the **New Password** field and retype the new password in the **Confirm New Password** field.

4 Click **Change Password**.

Your new password is saved.

Create or Load a System Profile

The Profile Editor in the Control Panel enables you to change device settings and update software. You can also save profiles onto your computer and load a profile to your CX5500 system.

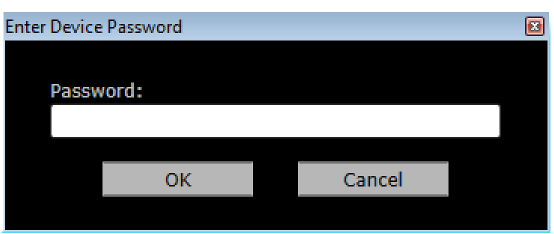
To create a profile:

1 On your computer, start the **CX5100/CX5500 Control Panel** application.

The Control Panel opens and your device's system information displays in the **System** tab.

2 In the **Control Panel**, click **Profile Editor**.

The Enter Device Password dialog displays.



3 Enter the **Device Password** and click **OK**.

Note that the default device password is the system's serial number (see figure Location of the Serial Number Label on the Power Data Box for the location of the system's serial number).

4 On the **Software Update** tab, enter the name of the **Update Server** and select values for the **Update Frequency** and **Update Time** fields.

5 On the **Advanced** tab, select options for the following settings:

  a Choose the **Mute Button Function**. Select **Microphone only** to mute the audio only or select **Microphone and Camera** to mute the audio and video when you touch the Mute button.

  b Select the **Power Frequency** for your system.

  c Choose the **USB Connectivity Reset Interval** and the **USB Connectivity Reset Time**.

6 Do one of the following:

  - Click **Apply to Device** to save the profile on your CX5500 system.
  - Click **Save to File (PC)** to save the profile to your computer. Specify the name of the file and the location of where to save the profile and click **Save**.

You can also load a profile from the device, a saved profile from your computer, or a default system profile onto your CX5500 systems.

To load a profile:

1 In the **Profile Editor** tab, click **Load Profile**.

2 Select one of the following options:

  - **Load from Device** Uploads the profile saved on the system.
  - **Load from File (PC)** Uploads a profile saved on your computer on to the system.
  - **Load Default Profile** Uploads the factory default profile for the system.

3 After you make your selection, click **Apply to Device**.

  The profile is saved onto the CX5500 system.

Update the CX5500 System’s Software Automatically

You can configure your system to check for available updates automatically, or you can update the software for your CX5500 system manually in the Control Panel or upload new software to the system using a USB flash drive.

To update the software automatically:

1 In the **Profile Editor** tab, select **Software Update**.

2 Enter the name of the **Update Server**.

3 Select how often your system updates for **Update Frequency**.

4 Select what time your system updates for **Update Time**.

  Your CX5500 system retrieves software updates from the server on the chosen date and time, if available.

To update the software manually:

1 Click **System > Software Update**.

2 Click **Update Now** to start the update.

  The system uploads the software update from the server, if available.

Troubleshoot Your CX5500 System

This section shows you some tools and techniques for troubleshooting the CX5500 system running Polycom® UC Software. The phone can provide feedback in the form of on-screen error messages, status indicators, and log files for troubleshooting issues.

This section includes information on:

- Understand Error Message Types
- Status Menu
- Log Files
- Manage the Phone’s Memory Resources
- Test Phone Hardware
- Upload a Phone’s Configuration
- Network Diagnostics
- Ports Used on the CX5500 System

This section also addresses phone issues, likely causes, and corrective actions. Issues are grouped as follows:

- Power and Startup Issues
- Dial Pad Issues
- Screen and System Access Issues
- Calling Issues
- Display Issues
- Audio Issues
- Licensed Feature Issues
- Upgrading Issues
- SoundStation Duo Failover Issues

Review the latest *UC Software Release Notes* on the Polycom UC Software Support Center for known problems and possible workarounds. If a problem is not listed in this section or in the latest *Release Notes*, contact your Certified Polycom Reseller for support.

Understand Error Message Types

Several types of errors can occur while the phone is booting. If an error occurs, the phone will inform you by displaying an error message. Errors can affect how the phone boots up. If the error is fatal, the phone will not be able to boot until the error is resolved. If the error is recoverable, the phone will continue to boot but the phone’s configuration may change.

Error Messages

Most of the following errors will be logged to the phone’s boot log. However, if you are having trouble connecting to the provisioning server, the phone will likely not be able to upload the boot log.

Failed to get boot parameters via DHCP

The phone does not have an IP address and therefore cannot boot. Check that all cables are connected, the DHCP server is running, and that the phone has not been set to a VLAN that is different from the DHCP server. Check the DHCP configuration.

Could not contact boot server using existing configuration

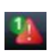
The phone could not contact the provisioning server, but the causes may be numerous. It may be cabling issue, it may be related to DHCP configuration, or it could be a problem with the provisioning server itself. The phone can recover from this error so long as it previously downloaded a valid application BootROM image and all of the necessary configuration files.

Error, application is not present!

This message indicates that the phone has no application stored in device settings, that the phone could not download an application, and that the phone cannot boot. To resolve this issue, you must download compatible Polycom UC Software to the phone using one of the supported provisioning protocols. You need to resolve the issue of connecting the phone to the provisioning server and provide a compatible software image on the provisioning server. This error is fatal, but recoverable.

Polycom UC Software Error Messages

The warning notification feature provides users a visual indication that one or more error conditions exist. When the warning notification displays, users will see:

- An informative message when the warning is first detected
- A warning icon  displays in the status bar
- A persistent list of current warnings, which can be viewed from **Status** > **Diagnostics** > **Warnings**

Config file error: Files contain invalid params: <filename1>, <filename2>,...
Config file error: <filename> contains invalid params.
The following contain pre-3.3.0 params: <filename>

These messages display if any of the following parameters are found in the configuration files:

- tone.chord.ringer.x.freq.x
- se.pat.callProg.x.name
- ind.anim.IP_500.x.frame.x.duration
- ind.pattern.x.step.x.state
- feature.2.name
- feature.9.name

This message also appears if any configuration file contains:

- More than 100 unknown parameters, or

- More than 100 out-of-range values, or
- More than 100 invalid values.

To update the configuration files to use the correct parameters, see Change Configuration Parameter Values for details.

Line: Unregistered

This message displays if a line fails to register with the call server.

Login credentials have failed. Please update them if information is incorrect.

This message displays when the user enters incorrect login credentials (**Status** > **Basic** > **Login Credentials**).

Missing files, config. reverted

This message displays when errors in the configuration and a failure to download the configuration files force the phone to revert to its previous (known) condition with a complete set of configuration files. This will also display if the files listed in the ***<MAC Address>*.cfg** file are not present on the provisioning server.

Network Authentication Failure

This message displays if 802.1X authentication with the CX5500 system fails. The error codes shown in the table Event Codes and Descriptions display on the phone’s screen—if the **Details** soft key is selected—and in the log files:

Event Codes and Descriptions

Event Code	*Description*	*Comments*
1	Unknown events	This includes any event listed in this table.
2	Mismatch in EAP Method type Authenticating server's list of EAP methods does not match with clients’.	
30xxx	TLS Certificate failure The TLS certificate-related failures. "xxx" when having a non-zero value, is the standard TLS alert message code. For example, if a bad/invalid certificate (on the basis of its signature and/or content) is presented by the phone, "xxx" will be 042. If the exact reason for the certificate being invalid is not known, then the generic certificate error code will be xxx=000.	See section 7.2 of RFC 2246 for further TLS alert codes and error codes.

Event Code	*Description*	*Comments*
31xxx	Server Certificate failure Certificate presented by the server is considered invalid. "xxx" can take the following values: • 009 - Certificate not yet Valid • 010 - Certificate Expired • 011 - Certificate Revocation List (CRL) not yet Valid • 012 - CRL Expired	
4xxx	Other TLS failures This is due to TLS failure other than certification related errors. The reason code (the TLS alert message code) is represented by "xxx". For example, if the protocol version presented by the server is not supported by the phone, then xxx will be 70, and the EAP error code will be 4070.	See section 7.2 of RFC 2246 for further TLS alert codes and error codes.

Network link is down

Link failures are indicated with the message 'Network link is down'. This message displays on the screen whenever the phone is not in the menu system and persists until the link problem is resolved. Call related functions and the soft keys and line keys are disabled when the network is down; however the menu works.

Status Menu

Debugging of a single phone may be possible by examining the phone's status menu. Tap **Settings > Status** to view the Status menu. Tap one of the Status menu items to view that item. Each of the menu items is explained next.

Under the **Platform** menu, you can get details on the phone's serial number or MAC address, the current IP address, the application version, the name of the configuration files in use, and the address of the provisioning server.

In the **Network** menu, you can find information about the TCP/IP Setting, Ethernet port speed, connectivity status of the PC port (if it exists), and statistics on packets sent and received since last boot. You can also find out the last time the phone rebooted. The **Call Statistics** screen shows packets sent and received on the last call.

The **Lines** menu shows you details about the status of each line that has been configured on the phone.

The **Diagnostics** menu offers a series of hardware tests to verify correct operation of the microphone, speaker, and touchscreen In addition to the hardware tests, the Diagnostics menu has a series of real-time graphs for CPU, network, and memory use that can be helpful for diagnosing performance issues.

Log Files

The CX5500 system logs various events to files stored in the flash file system and periodically uploads these log files to the provisioning server. The files are stored in the phone’s home directory or a user-configurable directory. You can also configure a phone to send log messages to a syslog server.

There is one log file for the UC Software. When a phone uploads its log files, the files are saved on the provisioning server with the MAC address of the phone prepended to the file name. For example, **0004f200360b-app.log** is the file associated with MAC address 00f4f200360b. The application log file is uploaded periodically or when the local copy reaches a predetermined size. For more information on log file contents, see the reference section <log/>.

The amount of logging that the phone performs can be tuned for the application to provide more or less detail on specific components of the phone’s software. For example, if you are troubleshooting a SIP signaling issue, you are not likely interested in DSP events. Logging levels are adjusted in the configuration files or via the Web Configuration Utility. You should not modify the default logging levels unless directed to by Polycom Customer Support. Inappropriate logging levels can cause performance issues on the phone.

In addition to logging events, the phone can be configured to automatically execute command-line instructions at specified intervals that output run-time information such as memory utilization, task status, or network buffer contents to the log file. These techniques should only be used in consultation with Polycom Customer Support.

Logging Options

Each of the components of the Polycom UC software is capable of logging events of different severity. This allows you to capture lower severity events in one part of the application, and high severity events for other components.

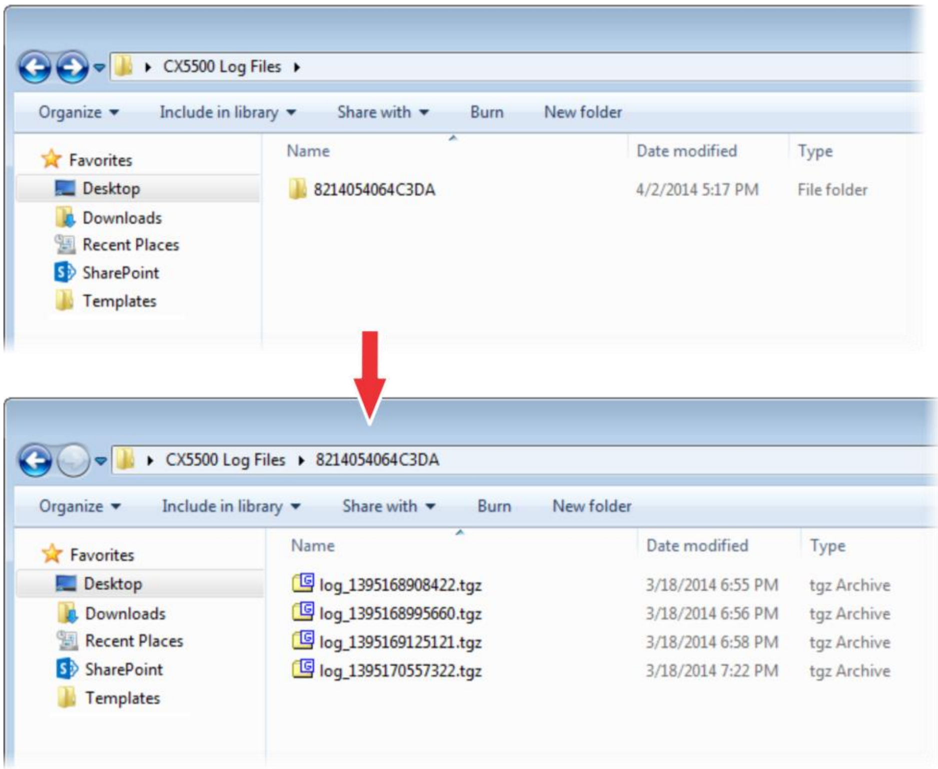
The following are two options for retrieving system log files for the CX5500 system:

- USB drive
- Log level parameter settings

When you connect a USB flash drive to the CX5500 system, the system creates a new folder on the flash drive. The folder is named with the device’s serial number and the system’s log files are saved as .tar files in the device folder.

The following figure shows an example device folder with log files.

Device Folder and Log Files for the CX5500 System



To retrieve system log files using a USB drive:

» Connect a USB flash drive to the USB port on the tabletop unit or on the power data box. Make sure there is no software update package on the flash drive.

The logs are transferred automatically. Note that it takes approximately one minute to complete the transfer.

The parameters for log level settings are found in the **techsupport.cfg** configuration file. They are `log.level.change.module_name`. Log levels range from 1 to 6 (1 for the most detailed logging, 6 for critical errors only). There are many different log types that can be adjusted to assist with the investigation of different problems. The exact number of log types is dependent on the phone model.

When testing is complete, remember to remove the configuration parameter from the configuration files.

There are other logging parameters, describe next, that you may wish to modify. Changing these parameters will not have the same impact as changing the logging levels, but you should still understand how your changes will affect the phone and the network.

- `log.render.level`—Sets the lowest level that can be logged (default=1)
- `log.render.file.size`—Maximum size before log file is uploaded (default=32 kb)
- `log.render.file.upload.period`—Frequency of log uploads (default is 172800 seconds = 48 hours)
- `log.render.file.upload.append`—Controls whether log files on the provisioning server are overwritten or appended, not supported by all servers (default=1 so files are appended)

- `log.render.file.upload.append.sizeLimit`—Controls the maximum size of log files on the provisioning server (default=512 kb)
- `log.render.file.upload.append.limitMode`—Control whether to stop or delete logging when the server log reaches its maximum size (default=delete)

Scheduled Logging

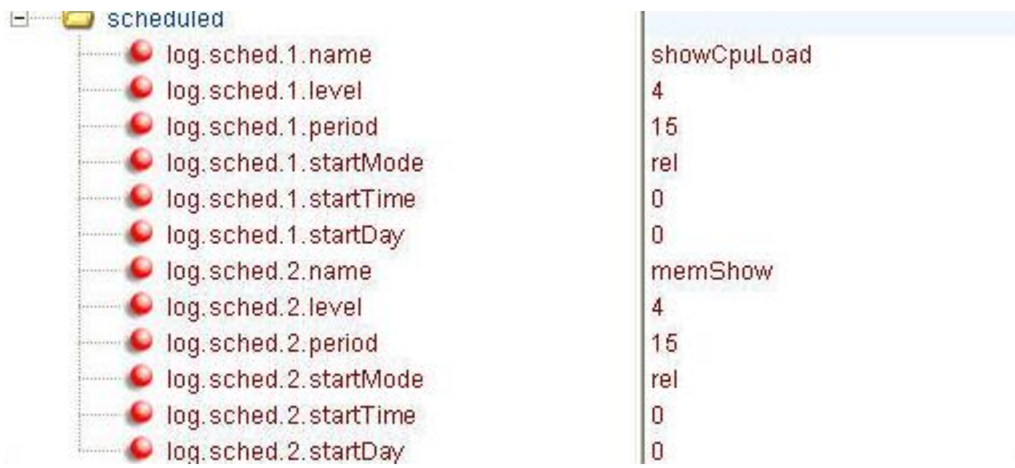
Schedules logging is a powerful tool that can help you troubleshoot issues that occur after the phone has been operating for some time.

The output of these instructions is written to the application log, and can be examined later (for trend data).

The parameters for scheduled logging are found in the **techsupport.cfg** configuration file. They are `log.sched.module_name`. Note that passwords display in a level 1 .cfg log file.

See the following figure for an example of a configuration file and the resulting log file.

Scheduled Logging Log File



```
0522163033|slog |4|01|#############################################
0522163033|slog |4|01|Running showCpuLoad
0522163033|slog |4|01|Cpu load is 6.0%, and the average is 57.6%.
0522163033|slog |4|01|#############################################
0522163033|slog |4|01|#############################################
0522163033|slog |4|01|Running memShow
0522163033|slog |4|01| status    bytes     blocks   avg block  max block
0522163033|slog |4|01| ------ ---------- --------- ---------- ----------
0522163033|slog |4|01|current
0522163033|slog |4|01|   free    9410608        65     144778    9257824
0522163033|slog |4|01|  alloc   11147888     31569        353          -
0522163033|slog |4|01|cumulative
0522163033|slog |4|01|  alloc   18961376     58186        325          -
0522163033|slog |4|01|#############################################
0522163048|slog |4|01|#############################################
0522163048|slog |4|01|Running showCpuLoad
0522163048|slog |4|01|Cpu load is 6.0%, and the average is 47.1%.
0522163048|slog |4|01|#############################################
0522163048|slog |4|01|#############################################
0522163048|slog |4|01|Running memShow
0522163048|slog |4|01| status    bytes     blocks   avg block  max block
0522163048|slog |4|01| ------ ---------- --------- ---------- ----------
```

Reading a Boot Log File

See the following figure for an example of a boot log file:

Boot Log File

```
0100000000|so   |4|00|+++ Note that bootrom log times are in GMT +++
0100000000|cfg  |4|00|Initial log entry
0100000000|copy |3|00|Initial log entry
0100000000|hw   |4|00|Initial log entry.
0100000000|ethf |4|00|Initial log entry.
0522182911|wdog |4|00|Initial log entry
0522182911|cdp  |3|00|CDP is DISABLED.
0522182911|so   |3|00|Platform: Model=SoundPoint IP 450, Assembly=2345-12450-001 Rev=3
0522182911|so   |3|00|Platform: Board=2345-12450-001 2
0522182911|so   |3|00|Platform: MAC=0004f21db094, IP=Resolving, Subnet Mask=Resolving
0522182911|so   |3|00|Platform: BootBlock=2.8.1 (12450_001) 04-Jun-08 17:04
0522182911|so   |3|00|Application, main: Label=BOOT, Version=4.1.2.0009 20-Jul-08 21:57
0522182911|so   |3|00|Application, main: P/N=3150-11069-412
0522182911|app1 |4|00|Initial log entry.
0522182912|so   |3|00|Link status is Net up Speed 100 full Duplex, PC down.
0522182916|cdp  |3|00|CDP received a response from a switch. CDP enabled.
0522182916|cdp  |3|00|Native VLAN Id is 1
0522182916|cdp  |3|00|No Auxiliary VLAN found
```

The following figure shows a number of boot failure messages:

Boot Failure Messages

```
0522183251|cfg  |3|00|Beginning to provision phone
0522183251|copy |3|00|'ftp://plcmspip:****@172.23.2.92/2345-12450-001.bootrom.ld' from
0522183251|copy |4|00|Download of '2345-12450-001.bootrom.ld' FAILED on attempt 1 (addr
0522183251|copy |4|00|Server '172.23.2.92' said '2345-12450-001.bootrom.ld' is not pres
0522183251|cfg  |4|00|Could not get all 512 bytes of the header
0522183251|copy |3|00|'ftp://plcmspip:****@172.23.2.92/bootrom.ld' from '172.23.2.92'
0522183251|copy |4|00|Download of 'bootrom.ld' FAILED on attempt 1 (addr 1 of 1)
0522183251|copy |4|00|Server '172.23.2.92' said 'bootrom.ld' is not present
0522183251|cfg  |4|00|Could not get all 512 bytes of the header
0522183251|cfg  |3|00|bootROM file not present on boot server
0522183251|copy |3|00|'ftp://plcmspip:****@172.23.2.92/0004f21db094.cfg' from '172.23.2
0522183251|copy |4|00|Download of '0004f21db094.cfg' FAILED on attempt 1 (addr 1 of 1)
0522183251|copy |4|00|Server '172.23.2.92' said '0004f21db094.cfg' is not present
0522183251|copy |3|00|Update of '/ffs0/init.mac' failed, leaving local copy intact
0522183251|copy |3|00|'ftp://plcmspip:****@172.23.2.92/000000000000.cfg' from '172.23.2
0522183251|copy |3|00|Download of '000000000000.cfg' succeeded on attempt 1 (addr 1 of
```

Reading an Application Log File

The following figure shows portions of an application log file:

Application Log File

```
0522184554|log  |*|01|Initial log entry. Current logging level 4
0522184554|so   |*|01|Initial log entry. Current logging level 3
0522184554|so   |*|01|---------- Initial log entry ----------
0522184554|so   |*|01|Platform: Model=SoundPoint IP 450, Assembly=2345-12450-001 Rev=
0522184554|so   |*|01|Platform: MAC=0004f21db094, IP=172.23.61.141, Subnet Mask=255.2
0522184554|so   |*|01|Platform: BootBlock=2.8.1 (12450_001) 04-Jun-08 17:04
0522184554|so   |*|01|Platform: Bootrom=4.1.2.0009 20-Jul-08 21:57
0522184554|so   |*|01|Application, main: Label=SIP, Version=3.1.3.0439 26-Apr-09 23:5
0522184554|so   |*|01|Application, main: P/N=3150-11530-313
0522184554|wdog |*|01|Initial log entry. Current logging level 4
0522184554|ethf |*|01|Initial log entry. Current logging level 4
0522184554|so   |5|01|utilCertificateInit failed.
0522184554|hw   |*|01|Initial log entry. Current logging level 4
0522184554|ares |*|01|Initial log entry. Current logging level 4
0522184554|dns  |*|01|Initial log entry. Current logging level 3
0522184554|cfg  |*|01|Initial log entry. Current logging level 3

0522114602|so   |*|01|System Info Reports:
0522114602|so   |*|01|  CPU is TNETV1055/C55x, rev 2 running at 150MHz with memory at
0522114602|so   |*|01|  Board is identified as PolycomSoundPointIP-SPIP_450.
0522114602|so   |*|01|  DRAM_LO: 0x94000000.  DRAM_SIZE: 32 MB
0522114602|so   |*|01|  Clocks are VBUSP: 125MHz, VBUS: 75MHz, USB: 25MHz, LCD: 20MHz,
0522114602|key  |*|01|Initial log entry. Current logging level 4
0522114602|ht   |*|01|Initial log entry. Current logging level 4
0522114602|httpd|*|01|Initial log entry. Current logging level 4
0522114602|ssps |*|01|Application, comp. 1: Label=PolyDSP Titan Mem1 FS5 (G.729), Vers
```

```
0522185324|cfg  |3|01|Prm|Check of configuration files suceeded
0522185324|cfg  |3|01|Prm|Phone successfully provisioned
0522185324|cfg  |*|01|Prm|Configuration file "001-phone1.cfg" is from template phone1
0522185324|cfg  |*|01|Prm|Configuration file "001-phone1.cfg" SHA1 digest: B712DCCA39
0522185324|cfg  |*|01|Prm|Configuration file "001-sip.cfg" is from template sip.cfg,
0522185324|cfg  |*|01|Prm|Configuration file "001-sip.cfg" SHA1 digest: B4E4534529797
0522185324|so   |3|01|Success provisioning.
```

```
0522120608|ldap |*|01|Initial log entry. Current logging level 4
0522120608|ldap |4|01|ldap: Not Enabled
0522120608|ldap |4|01|cDynamicData::cDynamicData:cDynamicData:Failed
0522120608|efk  |*|01|Initial log entry. Current logging level 4
0522120608|so   |*|01|[SoNcasC]: App-Ctx (6045551234) [0-6045551234]
0522120608|sip  |4|01|NAPTR query for host 'as-test' returned no results
0522120608|app1 |*|01|[InitializeBacklightIntensity] m_nDefaultMin = 0, m_nDefaultLow
0522120608|sip  |4|01|Registration failed User: 6045551234, Error Code:404 Not Found
0522120608|cfg  |4|01|Edit|Error 0x380003 attempting stat of /ffs0/local/0004f21db094-
```

Reading a Syslog File

The figure Syslog File shows a portion of a syslog log file. Note that the messages look identical to the normal log except for the addition of a timestamp and IP address:

Syslog File

```
Jan  0 00:00:00 172.23.7.249 0100000000|so   |4|00|---------- Initial log entry ----------
Jan  0 00:00:00 172.23.7.249 0100000000|so   |4|00|+++ Note that bootrom log times are in GMT +++
Jan  0 00:00:00 172.23.7.249 0100000000|cfg  |4|00|Initial log entry
Jan  0 00:00:00 172.23.7.249 0100000000|copy |3|00|Initial log entry
Jan  0 00:00:00 172.23.7.249 0100000000|hw   |4|00|Initial log entry.
Jan  0 00:00:00 172.23.7.249 0100000000|ethf |4|00|Initial log entry.
Feb 13 01:12:39 172.23.7.249 0213011239|wdog |4|00|Initial log entry
Feb 13 01:12:39 172.23.7.249 0213011239|cdp  |3|00|CDP is DISABLED.
Feb 13 01:12:39 172.23.7.249 0213011239|so   |3|00|Platform: Model=SoundPoint IP 650, Assembly=2345-126
Feb 13 01:12:39 172.23.7.249 0213011239|so   |3|00|Platform: Board=2345-12600-001 1
Feb 13 01:12:39 172.23.7.249 0213011239|so   |3|00|Platform: MAC=0004f2111511, IP=Resolving, Subnet Mas
Feb 13 01:12:39 172.23.7.249 0213011239|so   |3|00|Platform: BootBlock=2.7.0 (12600_001) 30-May-06 15:5
Feb 13 01:12:39 172.23.7.249 0213011239|so   |3|00|Application, main: Label=BOOT, Version=4.1.0.0219 10
Feb 13 01:12:39 172.23.7.249 0213011239|so   |3|00|Application, main: P/N=3150-11069-410
Feb 13 01:12:39 172.23.7.249 0213011239|appl |4|00|Initial log entry.
Feb 13 01:12:40 172.23.7.249 0213011240|so   |3|00|Link status is Net down, PC down.
Feb 13 01:12:41 172.23.7.249 0213011241|so   |3|00|Link status is Net up Speed 100 half Duplex, PC down
Feb 13 01:12:41 172.23.7.249 0213011241|cdp  |3|00|CDP is disabled.
Feb 13 01:12:45 172.23.7.249 0213011245|appl |3|00|DNS resolver servers are '172.23.0.200' '172.23.0.23
Feb 13 01:12:45 172.23.7.249 0213011245|appl |3|00|DNS resolver search domain is 'vancouver.polycom.com
Feb 13 01:12:45 172.23.7.249 0213011245|appl |3|00|Bootline: esw(3,0)bootHost:flash e=172.23.7.249:fff
Apr 15 22:32:22 172.23.7.249 0415223222|appl |3|00|Time has been set from 172.23.0.200 (172.23.0.200).
Apr 15 22:32:22 172.23.7.249 0415223222|appl |3|00|DHCP returned result 0x3E7 from server 172.23.0.232.
Apr 15 22:32:22 172.23.7.249 0415223222|appl |3|00|   Phone IP address is 172.23.7.249.
Apr 15 22:32:22 172.23.7.249 0415223222|appl |3|00|   Subnet mask is 255.255.0.0.
Apr 15 22:32:22 172.23.7.249 0415223222|appl |3|00|   Gateway address is 172.23.2.240.
Apr 15 22:32:22 172.23.7.249 0415223222|appl |3|00|   Time server is 172.23.0.200.
Apr 15 22:32:22 172.23.7.249 0415223222|appl |3|00|   GMT offset is -28800 seconds.
```

Web Info: Using Syslog on Polycom Phones

For more information about syslog, see *Feature Profile 17124: Using Syslog on Polycom Phones*.

Manage the CX5500 System's Memory Resources

The CX5500 system is designed to operate optimally in a variety of deployments and real-world environments. Each new software release adds new features and capabilities that require varying degrees of the system's memory resources. To ensure your systems and their configured features operate smoothly, you need to check that the systems have adequate available memory resources. If you are using a range of phone features—especially customized or advanced features—you may need to manage phone memory resources. To help you optimize your CX5500 system's features and memory resources, Polycom provides several tools and troubleshooting tips.

Identify Symptoms

When the phone memory resources start to run low, you may notice one or more of the following symptoms:

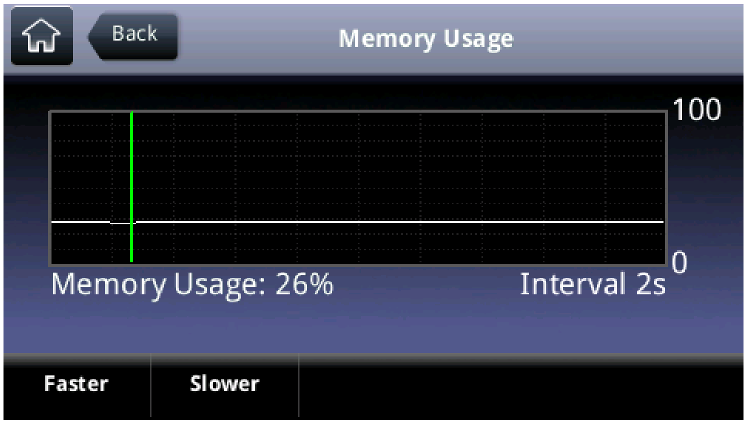
- The phones reboot or freeze up.
- The phones do not download all ringtones, directory entries, backgrounds, or XML dictionary files.
- Applications running in the microbrowser or browser stop or do not run at all.

The next sections show you how to check your phone's available memory and manage the phone features to make phone memory available.

Check the Phone's Available Memory

You can use two methods to quickly check whether you need to manage your phone's memory. Before you begin checking, load and configure the features and files you want to make available on the phone.

Using the first method, on your phone's keypad or touch pad interface, choose **Status > Diagnostics > Graphs > Memory Usage** as shown next.



Use the *Memory Usage* chart to check what the current Memory Usage amount is. Typically, you want to ensure that the phone is running at less than 95 percent of its available memory.

If the phone is using more than 95 percent of its available memory, you may need to take steps to reduce this amount. **Error! Reference source not found.**

The second method you can use to confirm whether you need to manage your phone's memory is to check the app log files. The app log file is enabled by default and is saved to your provisioning server directory with the MAC address of the phone prepended to the app log file. For example, if the MAC address of your phone is **0004f2203b0**, the app log file name will be **0004f2203b0-app.log**.

Open the app log. If you see the message shown next in the following figure, you need to manage your phone's memory resources.

Application Log Error Message

```
000014.458|dns   |*|00|DNS resolver servers are '172.23.0.200' '172.23.0.239'
000014.458|dns   |*|00|DNS resolver search domain is 'vancouver.polycom.com'
000014.460|cfg   |*|00|RT|Primary IP changed to 172.23.70.29 subnet mask 255.255.0.0
000016.412|ib    |*|00|Initial log entry. Current logging level 4
000016.414|so    |*|00|Network initialized. Starting network tasks.
000016.428|cfg   |5|00|Prm|Parameter lcl.ml.lang requested type 2 but is of type 4
000016.428|cfg   |5|00|Prm|Type 2 4 0 for parameter lcl.ml.lang is not valid
000016.658|sip   |*|00|Fast Boot Measurement Point: Ready for Call, uptime: 16.658 sec.
000016.982|tr69  |*|00|Initial log entry. Current logging level 4
000016.984|cfg   |*|00|Prov|Starting to update 2345-12670-001.sip.ld
000016.994|app1  |*|00|Ctx [1] Registered [true]
000017.004|res   |4|00|[ResFinderC]: Minimum free memory reached. 0xaf150.
000017.012|cfg   |*|00|Prov|Finished updating configuration.|
```

Web Info: Reading the App Log Files

For more information on reading the log files see the section Log Files.

Test Phone Hardware

You can view diagnostic information from the **Diagnostics** menu on your phone (**Settings > Status > Diagnostics**).

If you select **Diagnostics > Test Hardware**, you can select one of the following menu items to perform a hardware diagnostic test:

- **Audio Diagnostics** Test the speaker and microphones.
- **Display Diagnostics** Test the LCD for faulty pixels.
- **Touch Screen Diagnostics** Test the touch screen response.

USB keyboards and mice are not supported.

Avoid connecting an external keyboard or mouse to the system.

Upload a Phone's Configuration

As of Polycom UC Software 3.3.0, you can upload the files representing a phone's current configuration. A number of files can be uploaded to the provisioning server, one for every active source as well as the current non-default configuration set.

As of Polycom UC Software 4.0.0, you can upload the phone's configuration through the Web Configuration Utility.

This is primarily a diagnostics tool to help find configuration errors.

To upload the phone's current configuration:

1. Navigate to the Upload Configuration menu on the phone (**Settings > Advanced > Admin Settings > Upload Configuration**).
2. Choose to upload the configuration from one of **All Sources**, **Configuration Files**, **Local**, or **Web**.

   You can select **Device Settings** if you perform this task using the Web Configuration Utility.
3. Press the **Upload** soft key.

   The phone uploads the configuration file to the location that you specify in prov.configUploadPath. For example, if you select **All Sources**, a file **<*MACaddress*>-update-all.cfg** is uploaded.

Network Diagnostics

In Polycom UC Software 4.0.0, ping and traceroute are added to the phone's diagnostics tools. These diagnostics can be used for troubleshooting network connectivity problems in the wired and wireless worlds.

Both tools are accessible by tapping **Settings** and selecting **Status > Diagnostics > Network**.

Enter a URL address (for example, http://www.google.com) or any IP address (for example, the system IP address or any other phone's IP address), and tap the **Enter** soft key.

Ports Used on the CX5500 System

See the table Ports Used by the CX5500 System for a list of the ports currently used by the Polycom UC Software.

Ports used by the CX5500 system

Port Number	*Protocol*	*Outgoing*	*Incoming*	*UDP or TCP*
21	FTP	Provisioning, Logs		TCP
22	SSH	Admin	Admin	TCP
53	DNS			UDP
67	DHCP	Server		UDP

Port Number	*Protocol*	*Outgoing*	*Incoming*	*UDP or TCP*
68	DHCP	Client		UDP
69	TFTP	Provisioning, Logs		UDP
80	HTTP	Provisioning, Logs, Pull Web interface, Poll		TCP
123	NTP	Time Server		UDP
389	LDAP			
443	HTTPS	Provisioning, Logs	HTTP Pull Web interface, HTTP Push	TCP
514	Syslog	Logs		
636	LDAP			
1023	Telnet	Admin		TCP
2222	RTP[2]	Media Packets	Media Packets	
2223	RTCP[2]	Media Packet Statistics	Media Packet Statistics	
5060	SIP	SIP signaling	SIP signaling	
5061	SIP over TLS	Secure signaling	Secure signaling	

1 Telnet is disabled by default.

2 RTP and RTCP can use any even port between 2222 and 2269, but this is configurable by setting `tcpIpApp.port.rtp.mediaPortRangeStart`.

Power and Startup Issues

The table Troubleshooting Power and Startup Issues describes possible solutions to several power and startup issues.

Troubleshooting Power and Startup Issues

The phone has power issues or the phone has no power.

Determine if the problem is caused by the phone, the AC outlet, or the PoE switch. Do one of the following:

- Verify that no lights appear on the unit when it is powered up.
- Check if the phone is properly plugged into a functional AC outlet.
- Make sure that the phone isn't plugged into an outlet controlled by a light switch that is off.
- If plugged into a power strip, try plugging directly into a wall outlet instead.

The phone will not boot.

If your phone will not boot, there may be a corrupt or invalid firmware image or configuration on the phone:

- Ensure that the provisioning server is accessible on the network and a valid software load and valid configuration files are available.
- Ensure that the phone is pointing to the provisioning server on the network.
- Reboot the phone.

Touch Screen Issues

The LCD touch screen menu includes a panel in which you can test the sensitivity of the touch screen. Navigate to **Settings > Status > Diagnostics >Test Hardware > Touch Screen Diagnostics** to test the touch screen.

Screen and System Access Issues

The table Troubleshooting Screen and System Access Issues describes possible solutions to screen and system access issues.

Troubleshooting Screen and System Access Issues

There is no response from feature key presses.

If your phone is not in the active state, do one of the following:

- Press the keys more slowly.
- Check to see whether or not the key has been mapped to a different function or disabled.
- Make a call to the phone to check for inbound call display and ringing. If successful, try to press feature keys while a call is active to access a Directory or Buddy Status, for example.
- Navigate to **Settings > Status > Lines** to confirm the line is actively registered to the call server.
- Reboot the phone to attempt re-registration to the call server (navigate to **Settings > Advanced > Reboot Phone**)

The display shows the message *Network Link is Down.*

If you see this message, the LAN cable is not properly connected. Do one of the following:

- Check termination at the switch or hub (furthest end of the cable from the phone).
- Check that the switch or hub is operational (flashing link/status lights).
- Press **Settings > Status > Network**. Scroll down to verify that the LAN is active.
- Ping the phone from another machine.
- Reboot the phone to attempt re-registration to the call server (navigate to **Settings > Advanced > Reboot Phone**).

Calling Issues

The table Troubleshooting Calling Issues provides possible solutions to a number of generic calling issues.

Troubleshooting Calling Issues

There is no dial tone.

If there is no dial tone, power may not be correctly supplied to the phone, try one of the following:

- Check that the display is illuminated.
- Make sure the LAN cable is inserted properly at the rear of the phone (try unplugging and re-inserting the cable).
- If using in-line powering, have your system administrator check that the switch is supplying power to the phone.

The phone does not ring.

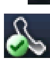
If there is a no ring tone, but the phone displays a visual indication when it receives an incoming call, do the following:

- Adjust the ring level from the front panel using the volume up/down keys.

The line icon shows an unregistered line icon.

If you see one of the following icons the phone line is unregistered. Register the line and try to place a call.

Unregistered Line Icon: 

Registered Line Icon: 

Display Issues

The table Troubleshooting Display Issues provides tips for resolving display screen issues.

Troubleshooting Display Issues

There is no display or the display is incorrect.

If there is no display, power may not be correctly supplied to the phone. Do one of the following:

- Check that the display is illuminated.
- Make sure the power is inserted properly in the power data box.
- If using Power over Ethernet (PoE) powering, check that the PoE switch is supplying power to the phone.

Use the screen capture feature to determine if the display on the phone is incorrect (see Capture the Phone's Current Screen).

The display is too dark or too light. The phone contrast may be set incorrectly. To adjust the contrast, do one of the following: • Adjust the contrast (Refer the phone's user guide). • Reboot the phone to obtain the default level of contrast. • Use the screen capture feature to see if the screen displays properly in the capture (see Capture the Phone's Current Screen).
The display is flickering. Certain types of older fluorescent lighting cause the display to flicker. If your phone is in an environment lit with fluorescent lighting, do one of the following: • Move the CX5500 system away from the lights. • Replace the lights.

Audio Issues

The table Troubleshooting Audio Issues describes possible solutions to audio issues.

Troubleshooting Audio Issues

There are audio or echo issues If you experience echo issues, see *Technical Bulletin 16249: Troubleshooting Audio and Echo Issues on SoundPoint IP Phones*.

Licensed Feature Issues

You need a license for XT9 support. You can check your licenses on the device by navigating to **Settings > Status > Licenses**.

Upgrading Issues

The table Troubleshooting Software Upgrading Issues describes several possible solutions to issues that may occur during or after a software upgrade.

Troubleshooting Software Upgrading Issues

Certain settings or features are not working as expected on the phone The phone's configuration may be incorrect or incompatible. Check for errors on the phone by navigating to **Settings > Status > Platform > Configuration**. If there are *Errors Found*, *Unknown Params,* or *Invalid values*, correct your configuration files and restart the phone.

The phone displays a *Config file error* message for 5-seconds after it boots up (see the following figure)

Pre-UC Software 3.3.0 configuration files are being used with UC Software 3.3.0. Specifically, the following parameters are in the configuration files:

- one.chord.ringer.x.freq.1
- se.pat.callProg.x.name
- ind.anim.IP_500.x.frame.x. duration
- ind.pattern.1.step.x.state
- feature.2.name
- feature.9.name

Also the configuration files contain:

- more than 100 "unknown" parameters
- more than 100 "out-of-range" parameters
- more than 100 "invalid" parameters

Correct the configuration files, remove the invalid parameters, and restart the phone.

When you are upgrading phone software using the Web Configuration Utility, the phone is unable to connect to the Polycom Hosted Server.

Occasionally, the phone is unable to connect to the Polycom Hosted Server because:

- The Polycom Hosted Server is temporarily unavailable.
- There isn't any software upgrade information for the phone to receive.
- The network configuration is preventing the phone from connecting to the Polycom Hosted Server.

Note: UC Software 4.0.0 does not support internet access for software upgrades through a Web proxy.

To troubleshoot the issue:

- Try upgrading your phone later.
- Verify that new software is available for your phone. To check, see the Polycom UC Software/Polycom SIP Software Release Matrix.
- Verify that your network's configuration will allow the phone to connect to http://downloads.polycom.com.

If the issue persists, try manually upgrading your phone's software. To upgrade phone software using this method, see Set Up the Provisioning Server.

Miscellaneous Maintenance Tasks

This section shows you how to maintain the Polycom® UC Software and includes the following topics:

- Trusted Certificate Authority List
- Encrypt Configuration Files
- Internal Key Functions
- Assign a VLAN ID Using DHCP
- Parse Vendor ID Information
- Product, Model, and Part Number Mapping
- Capture the Phone's Current Screen
- LLDP and Supported TLVs

Trusted Certificate Authority List

The phone trusts the following certificate authorities by default:

- AAA Certificate Services by COMODO
- ABAecom (sub., Am. Bankers Assn.) Root CA
- Add Trust Class1 CA Root by COMODO
- Add Trust External CA Root by COMODO
- Add Trust Public CA Root by COMODO
- Add Trust Qualified CA Root by COMODO
- ANX Network CA by DST
- American Express CA
- American Express Global CA
- BelSign Object Publishing CA
- BelSign Secure Server CA
- COMODO CA Limited
- COMODO Certificate Authority
- Deutsche Telekom AG Root CA
- Digital Signature Trust Co. Global CA 1
- Digital Signature Trust Co. Global CA 2
- Digital Signature Trust Co. Global CA 3
- Digital Signature Trust Co. Global CA 4
- Entrust Worldwide by DST
- Entrust.net Premium 2048 Secure Server CA
- Entrust.net Secure Personal CA
- Entrust.net Secure Server CA

- Equifax Premium CA
- Equifax Secure CA
- Equifax Secure eBusiness CA 1
- Equifax Secure eBusiness CA 2
- Equifax Secure Global eBusiness CA 1
- GeoTrust Primary Certification Authority
- GeoTrust Global CA
- GeoTrust Global CA 2
- GeoTrust Universal CA
- GeoTrust Universal CA 2
- GTE CyberTrust Global Root
- GTE CyberTrust Japan Root CA
- GTE CyberTrust Japan Secure Server CA
- GTE CyberTrust Root 2
- GTE CyberTrust Root 3
- GTE CyberTrust Root 4
- GTE CyberTrust Root 5
- GTE CyberTrust Root CA
- GlobalSign Partners CA
- GlobalSign Primary Class 1 CA
- GlobalSign Primary Class 2 CA
- GlobalSign Primary Class 3 CA
- GlobalSign Root CA
- Go Daddy Class 2 Certification Authority Root Certificate
- Go Daddy Class 2 Certification Authority Root Certificate – G2
- National Retail Federation by DST
- RSA 2048 v3 Root CA
- Secure Certificate Services by COMODO
- TC TrustCenter, Germany, Class 1 CA
- TC TrustCenter, Germany, Class 2 CA
- TC TrustCenter, Germany, Class 3 CA
- TC TrustCenter, Germany, Class 4 CA
- Thawte Personal Basic CA
- Thawte Personal Freemail CA
- Thawte Personal Premium CA
- Thawte Premium Server CA
- Thawte Server CA

- Thawte Universal CA Root
- Trusted Certificate Services by COMODO
- UTN-DATA Corp SGC by COMODO
- UTN-USER First-Client Authentication and Email by COMODO
- UTN-USER First-Hardware by COMODO
- UTN-USER First-Object by COMODO
- UPS Document Exchange by DST
- ValiCert Class 1 VA
- ValiCert Class 2 VA
- ValiCert Class 3 VA
- Verisign 2048 Root CA
- VeriSign Class 4 Primary CA
- Verisign Class 1 Public Primary Certification Authority
- Verisign Class 1 Public Primary Certification Authority - G2
- Verisign Class 1 Public Primary Certification Authority - G3
- Verisign Class 2 Public Primary Certification Authority
- Verisign Class 2 Public Primary Certification Authority - G2
- Verisign Class 2 Public Primary Certification Authority - G3
- Verisign Class 3 Public Primary Certification Authority
- Verisign Class 3 Public Primary Certification Authority - G2
- Verisign Class 3 Public Primary Certification Authority - G3
- Versign Class 3 Public Primary Certification Authority – G5
- Verisign Class 4 Public Primary Certification Authority - G2
- Verisign Class 4 Public Primary Certification Authority - G3
- Verisign/RSA Commercial CA
- Verisign/RSA Secure Server CA
- Windows Root Update by COMODO

Troubleshooting: My Certificate Authority is Not Listed

Polycom endeavors to maintain a built-in list of the most commonly used CA Certificates. Due to memory constraints, we cannot ensure a complete set of certificates. If you are using a certificate from a commercial Certificate Authority not in the list above, you may submit a Feature Request for Polycom to add your CA to the trusted list. At this point, you can use the Custom Certificate method to load your particular CA certificate into the phone. Refer to *Technical Bulletin 17877: Using Custom Certificates on Polycom Phones*.

Encrypt Configuration Files

The phone can recognize encrypted files. Phones can download encrypted files from the provisioning server and can encrypt files before uploading them to the provisioning server. There must be an encryption key on the phone to perform these operations. You can encrypt configuration files (excluding the master configuration file), contact directories, and configuration override files.

You can generate your own 32 hex-digit, 128 bit key or use Polycom's Software Development Kit (SDK) to generate a key and to encrypt and decrypt configuration files on a UNIX or Linux server. The SDK is distributed as source code that runs under the UNIX operating system.

Web Info: Using the SDK to Encrypt Configuration Files

To request the SDK and quickly install the generated key, see *Quick Tip 67442: When Encrypting Polycom UC Software Configuration Files*.

The SDK generates a random key and applies Advanced Encryption Standard (AES) 128 in Cipher Block Chaining (CBC) mode. For example, a key can look like this:

`Crypt=1;KeyDesc=companyNameKey1;Key=06a9214036b8a15b512e03d53412006;`

The `device.set`, `device.sec.configEncryption.key`, and `device.sec.configEncryption.key.set` configuration file parameters are used to set the key on the phone.

If the phone doesn't have a key, it must be downloaded to the phone in plain text (a potential security concern if not using HTTPS). If the phone already has a key, a new key can be downloaded to the phone encrypted using the old key.

Polycom recommends that you give each key a unique descriptive string in order to identify which key was used to encrypt a file. This makes provisioning server management easier.

After encrypting a configuration file, it is useful to rename the file to avoid confusing it with the original version, for example rename **site.cfg** to **site.enc**. However, the directory and override filenames cannot be changed in this manner.

Troubleshooting: My Phone Keeps Displaying an Error Message for My Encrypted File

If a phone downloads an encrypted file that it cannot decrypt, the action is logged, and an error message displays. The phone will continue to do this until the provisioning server provides an encrypted file that can be read, an unencrypted file, or the file is removed from the master configuration file list.

To check whether an encrypted file is the same as an unencrypted file:

1 Run the *configFileEncrypt* utility (available from Polycom Support) on the unencrypted file with the "-d" option. This shows the "digest" field.

2 Look at the encrypted file using text editor and check the first line that shows a "Digest=…." field. If the two fields are the same, then the encrypted and unencrypted file are the same.

For security purposes, you can change the key on the phones and the server from time to time.

To change a key on the phone:

1 Put all encrypted configuration files on the provisioning server to use the new key.

The phone may reboot multiple times.

The files on the server must be updated to the new key or they must be made available in unencrypted format. Updating to the new key requires decrypting the file with the old key, then encrypting it with the new key.

2 Put the new key into a configuration file that is in the list of files downloaded by the phone (specified in **000000000000.cfg** or ***<MACaddress>*.cfg**).

3 Use the `device.sec.configEncryption.key` parameter to specify the new key.

4 Provisioning the phone again so that it will download the new key. The phone will automatically reboot a second time to use the new key.

Note that configuration files, contact directory files and configuration override files may all need to be updated if they were already encrypted. In the case of configuration override files, they can be deleted from the provisioning server so that the phone will replace them when it successfully boots.

Internal Key Functions

A complete list of internal key functions for enhanced feature keys and hard key mappings is shown in the table Key Labels and Internal Functions.

The following guidelines should be noted:

- The ***Function*** value is case sensitive.
- Some functions are dependent on call state. Generally, if the soft key displays on a call screen, the soft key function is executable.
- CallPickup refers to the soft key function that provides the menu with separate soft keys for parked pickup, directed pickup, and group pickup.
- Some functions depend on the feature being enabled. For example, BuddyStatus and MyStatus require the presence feature to be enabled.
- The table below shows only Line1 to Line6 functions.

Key Labels and Internal Functions

Function	*Description*	*Notes*
ACDAvailable	ACD available from idle	
ACDLogin	Login to ACD	
ACDLogout	Log out of ACD	
ACDUnavailable	ACD unavailable from idle	
Answer	Answer	Call screen only
Applications	Main Browser	

Function	*Description*	*Notes*
BuddyStatus	Buddy Status	
CallList	Call Lists	
Conference	Begin a conference call	Call screen only
Delete	Delete	
Dialpad0	Dialpad 0	
Dialpad1	Dialpad 1	
Dialpad2	Dialpad 2	
Dialpad3	Dialpad 3	
Dialpad4	Dialpad 4	
Dialpad5	Dialpad 5	
Dialpad6	Dialpad 6	
Dialpad7	Dialpad 7	
Dialpad8	Dialpad 8	
Dialpad9	Dialpad 9	
DialpadPound	Dialpad pound sign	
DialpadStar	Dialpad star sign	
DialpadURL	Dial name	Call screen only
Directories	Directories	
DoNotDisturb	Do Not Disturb menu	
EnterRecord	Enter a call record	Call screen only
Exit	Exit existing menu	Menu only
Hold	Toggle hold	
Join	Join	Call screen only
Line1	Line Key 1	
Line2	Line Key 2	
Line3	Line Key 3	
Line4	Line Key 4	
Line5	Line Key 5	

Function	*Description*	*Notes*
Line6	Line Key 6	
LockPhone	Lock the phone	
Messages	Messages menu	
MicMute	Mute the microphone	
MyStatus	View my status	
NewCall	New call	Call screen only
Null	Do nothing	
Offline	Offline for presence	
Page	Group Paging	
QuickSetup	Quick Setup feature	Call screen only
Redial	Redial	Call screen only
Select	Select	
ServerACDAgentAvailable	serverACDAgentAvailable	
ServerACDAgentUnavailable	serverACDAgentUnavailable	
ServerACDSignIn	serverACDSignIn	
ServerACDSignOut	serverACDSignOut	
Setup	Settings menu	
Silence	RingerSilence	Call screen only
SoftKey1	SoftKey 1	
SoftKey2	SoftKey 2	
SoftKey3	SoftKey 3	
SoftKey4	SoftKey 4	
SpeedDial	SpeedDial	
Split	Split	Call screen only
Talk	Push-to-Talk	
Transfer	Transfer	Call screen only
VolDown	Set volume down	
VolUp	Set volume up	

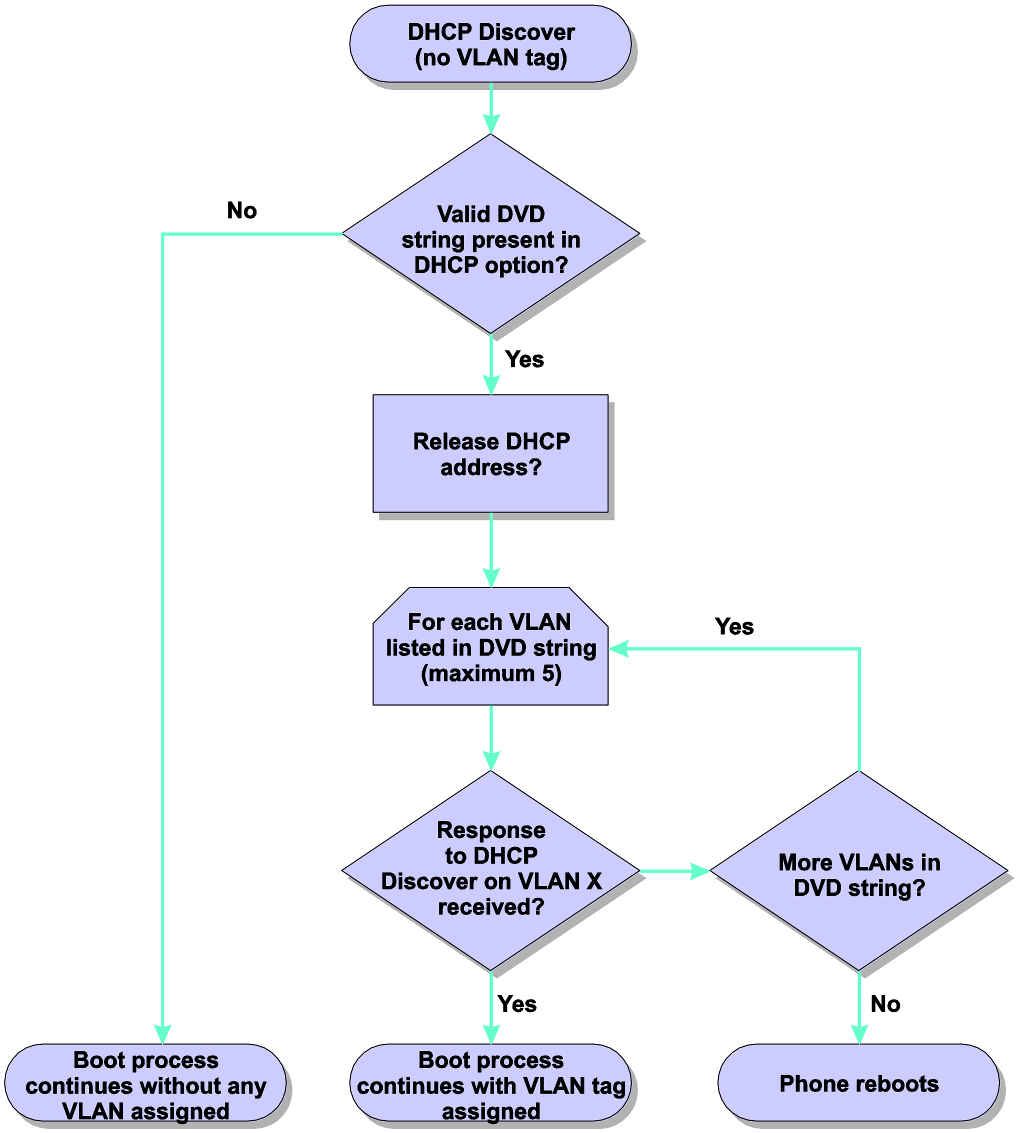
Assign a VLAN ID Using DHCP

In deployments where is not possible or desirable to assign a VLAN statically in the phone’s network configuration menu or use CDP (Cisco Discovery Protocol) or LLDP (Link-Layer Discovery Protocol) to assign a VLAN ID, it is possible to assign a VLAN ID to the phone by distributing the VLAN ID via DHCP.

When using this method to assign the phone’s VLAN ID, the phone first boots on the default VLAN (or statically configured VLAN, if first configured in the phone’s network configuration menu), obtains its intended VLAN ID from the DHCP offer, then continues booting (including a subsequent DHCP sequence) on the newly obtained VLAN.

See the figure VLAN Using DHCP Phone Boot Up Sequence for the phone boot-up sequence when assigning a VLAN ID via DHCP.

VLAN Using DHCP Phone Boot-Up Sequence



To assign a VLAN ID to a phone using DHCP:

» In the DHCP menu of the Main setup menu, set **VLAN Discovery** to **Fixed** or **Custom**.

- When set to Fixed, the phone will examine DHCP options 128,144, 157 and 191 (in that order) for a valid DVD string.
- When set to Custom, a value set in the **VLAN ID Option** will be examined for a valid DVD string.
  - DVD string in the DHCP option must meet the following conditions to be valid:
- Must start with "VLAN-A=" (case-sensitive)
- Must contain at least one valid ID
- VLAN IDs range from 0 to 4095
- Each VLAN ID must be separated by a "+" character

- The string must be terminated by a semi colon “;”
- All characters after the semi colon “;” will be ignored
- There must be no white space before the semi colon “;”
- VLAN IDs may be decimal, hex, or octal
  - The following DVD strings will result in the phone using VLAN 10:

```
VLAN-A=10;
VLAN-A=0x0a;
VLAN-A=012;
```

Note: VLAN Tags Assigned by CDP or LLDP

If a VLAN tag is assigned by CDP or LLDP, DHCP VLAN tags will be ignored.

Parse Vendor ID Information

After the phone boots, it sends a DHCP Discover packet to the DHCP server. This is found in the Bootstrap Protocol/option ‘Vendor Class Identifier’ section of the packet and includes the phone’s part number and the BootROM version. RFC 2132 does not specify the format of this option's data, and can be defined by each vendor. To be useful, every vendor's format must be distinguishable from every other vendor's format. To make our format uniquely identifiable, the format follows RFC 3925, which uses the IANA Private Enterprise number to determine which vendor's format should be used to decode the remaining data. The private enterprise number assigned to Polycom is 13885 (0x0000363D).

This vendor ID information is not a character string, but an array of binary data.

The steps for parsing are as follows:

1. Check for the Polycom signature at the start of the option:
   4 octet: 00 00 36 3d
2. Get the length of the entire list of sub-options:
   1 octet
3. Read the field code and length of the first sub-option, 1+1 octets
4. If this is a field you want to parse, save the data.
5. Skip to the start of the next sub-option.
6. Repeat steps 3 to 5 until you have all the data or you encounter the End-of-Suboptions code (0xFF).

For example, the following is a sample decode of a packet from an IP 601:

```
3c 74
```

- Option 60, length of Option data (part of the DHCP spec.)

```
00 00 36 3d
```

- Polycom signature (always 4 octets)

```
6f
```

- Length of Polycom data

```
01 07 50 6f 6c 79 63 6f 6d
```

- sub-option 1 (company), length, "Polycom"

```
02 15 53 6f 75 6e 64 50 6f 69 6e 74 49 50 2d 53 50 49 50 5f 36 30 31
```

- sub-option 2 (part), length, "CX5500"

```
03 10 32 33 34 35 2d 31 31 36 30 35 2d 30 30 31 2c 32
```

- sub-option 3 (part number), length, "2345-11605-001,2"

```
04 1c 53 49 50 2f 54 69 70 2e 58 58 58 58 2f 30 38 2d 4a 75 6e 2d 30 37 20 31
30 3a 34 34
```

- sub-option 4 (Application version), length, "SIP/Tip.XXXX/08-Jun-07 10:44"

```
05 1d 42 52 2f 33 2e 31 2e 30 2e 58 58 58 58 2f 32 38 2d 41 70 72 2d 30 35 20
31 33 3a 33 30
```

- sub-option 5 (BootROM version), length, "BR/3.1.0.XXXX/28-Apr-05

```
13:30"
ff
```

- end of sub-options

For the Updater, sub-option 4 and sub-option 5 will contain the same string. The string is formatted as follows:

```
<apptype>/<buildid>/<date+time>
where:
<apptype> can be 'BR' (BootROM) or 'SIP' (SIP Application)
```

Product, Model, and Part Number Mapping

You can use the master configuration file to direct phone upgrades to a software image and configuration files based on a phone model number, a firmware part number, or a phone's MAC address.

The part number has precedence over the model number, which has precedence over the original version.

For example, `CONFIG_FILES_2345-11560-001="phone1_2345-11560-001.cfg, sip_2345-11560-001.cfg"` will override `CONFIG_FILES_CX5500="phone1_CX5500.cfg, sip_CX5500.cfg"`, which will override `CONFIG_FILES="phone1.cfg, sip.cfg"` for the CX5500.

You can also add variables to the master configuration file that are replaced when the phone reboots. The variables include PHONE_MODEL, PHONE_PART_NUMBER, and PHONE_MAC_ADDRESS.

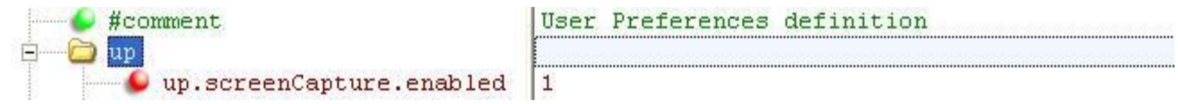
Capture the Phone's Current Screen

You can capture your phone's current screen using a Web browser.

To capture the phone's current screen:

1 Modify your configuration file to enable the screen capture feature.

2 Open your configuration file in an XML editor and add the following line:



3 Save the configuration file and update your phone's configuration.

4 On the phone, turn on the screen capture feature from the **Screen Capture** menu (**Settings > Basic > Preferences > Screen Capture**).

Turn the screen capture on again (repeat this step) each time the phone restarts or reboots.

5 In a Web browser, enter **http://<phoneIPaddress>/captureScreen** as the browser address.

To find your phone's IP address, navigate to **Settings > Status > Platform > Phone**.

The Web browser displays an image showing the phone's current screen. The image can be saved as a BMP or JPEG file.

LLDP and Supported TLVs

The Link Layer Discovery Protocol (LLDP) is a vendor-neutral Layer 2 protocol that allows a network device to advertise its identity and capabilities on the local network.

Web Info: Using the LLDP Protocol

The protocol was formally ratified as IEEE standard 802.1AB in May 2005. Refer to section 10.2.4.4 of the LLDP-MED standard.

The LLDP feature supports VLAN discovery and LLDP power management, but not power negotiation. LLDP has a higher priority than CDP and DHCP VLAN discovery.

Settings: Enabling VLAN Using Multiple Method

There are four ways to obtain VLAN on the phone and they can all be enabled, but the VLAN used is chosen by the priority of each method: 1. LLDP; 2. CDP; 3. DVD (VLAN Via DHCP); 4. Static (the VLAN ID is entered through the phone's user interface).

The following mandatory and optional Type Length Values (TLVs) are supported:

Mandatory:

- Chassis ID—Must be first TLV
- Port ID—Must be second TLV
- Time-to-live—Must be third TLV, set to 120 seconds

- End-of-LLDPDU—Must be last TLV
- LLDP-MED Capabilities
- LLDP-MED Network Policy—VLAN, L2 QoS, L3 QoS
- LLDP-MED Extended Power-Via-MDI TLV—Power Type, Power Source, Power Priority, Power Value

Optional:

- Port Description
- System Name—Administrator assigned name
- System Description—Includes device type, phone number, hardware version, and software version
- System Capabilities—Set as 'Telephone' capability
- MAC / PHY config status—Detects duplex mismatch
- Management Address—Used for network discovery
- LLDP-MED Location Identification—Location data formats: Co-ordinate, Civic Address, ECS ELIN
- LLDP-MED Inventory Management —Hardware Revision, Firmware Revision, Software Revision, Serial Number, Manufacturer's Name, Model Name, Asset ID

An LLDP frame shall contain all mandatory TLVs. The frame will be recognized as LLDP only if it contains mandatory TLVs. Polycom phones running the UC Software will support LLDP frames with both mandatory and optional TLVs. The basic structure of an LLDP frame and a table containing all TLVs along with each field is explained in Supported TLVs.

LLDP-MED Location Identification

As per section 10.2.4.4 of the LLDP-MED standard, LLDP-MED endpoint devices need to transmit Location Identification TLVs if they are capable of either automatically determining their physical location by use of GPS or radio beacon or capable of being statically configured with this information.

At present, the phones do not have the capability to determine their physical location automatically or provision to a statically configured location. Because of these limitations, the phones will not transmit Location Identification TLV in the LLDP frame. However, the location information from the switch is decoded and displayed on the phone's menu.

For more information on device configuration parameters, refer to the section <device/>.

Supported TLVs

The basic TLV format is as follows:

- TLV Type (7 bits) [0-6]
- TLV Length (9 bits) [7-15]
- TLV Information (0-511 bytes)

The following table lists the supported TLVs.

Supported TLVs

No	*Name*	*Type(7 bits) [0-6]*	*Length (9 bits) [7-15]*	*Type Length*	*Org. Unique Code (3 bytes)*	*Sub Type*
1	**Chassis-Id[1]**	**1**	**6**	**0x0206**	**-**	**5**
IP address of phone (4 bytes). Note that 0.0.0.0 is not sent until the phone has a valid IP address.						
2	**Port-Id[1]**	**2**	**7**	**0x0407**	**-**	**3**
MAC address of phone (6 bytes)						
3	**TTL**	**3**	**2**	**0x0602**	**-**	**-**
TTL value is 120/0 sec						
4	**Port description**	**4**	**1**	**0x0801**	**-**	**-**
Port description 1						
5	**System name**	**5**	**min len > 0, max len <= 255**	**-**	**-**	**-**
6	**System description**	**6**	**min len > 0, max len <= 255**	**-**	**-**	**-**
Manufacturer's name - "Polycom"; Refer to **Error! Reference source not found.**; Hardware version; Application ersion; BootROM version						
7	**Capabilities**	**7**	**4**	**0x0e04**	**-**	**-**
System Capabilities: Telephone and Bridge if the phone has PC port support and it is not disabled. Enabled Capabilities: Telephone and Bridge if phone has PC port support, it is not disabled and PC port is connected to PC.						
8	**Management Address**	**8**	**12**	**0x100c**	**-**	**-**
Address String Len - 5, IPV4 subtype, IP address, Interface subtype - "Unknown", Interface number - "0", ODI string Len - "0"						
9	**IEEE 802.3 MAC/PHY config/status[1]**	**127**	**9**	**0xfe09**	**0x00120f**	**1**
Auto Negotiation Supported - "1", enabled/disabled, Refer to PMD Advertise and Operational MAU.						
10	**LLDP-MED capabilities**	**127**	**7**	**0xfe07**	**0x0012bb**	**1**
Capabilities - 0x33 (LLDP-Med capabilities, Network policy, Extended Power Via MDI-PD, Inventory) Class Type III Note: Once support for configuring location Identification information is locally available: Capabilities - 0x37 (LLDP-Med capabilities, Network policy, Location Identification, Extended Power Via MDI-PD, Inventory) Class Type III						

No	*Name*	*Type(7 bits) [0-6]*	*Length (9 bits) [7-15]*	*Type Length*	*Org. Unique Code (3 bytes)*	*Sub Type*
11	**LLDP-MED network policy**[2]	**127**	**8**	**0xfe08**	**0x0012bb**	**2**
ApplicationType: Voice (1), Policy: (Unknown(=1)/Defined(=0) Unknown, if phone is in booting stage or if switch doesn't support network policy TLV. Defined, if phone is operational stage and Networkpolicy TLV is received from the switch.), Tagged/Untagged, VlanId, L2 priority and DSCP						
12	**LLDP-MED network policy**[2]	**127**	**8**	**0xfe08**	**0x0012bb**	**2**
ApplicationType: Voice Signaling (2), Policy: (Unknown(=1)/Defined(=0) Unknown, if phone is in booting stage or if switch doesn't support network policy TLV. Defined, if phone is operational stage and Networkpolicy TLV is received from the switch.), Tagged/Untagged, VlanId, L2 priority and DSCP. Note: Voice signaling TLV is sent only if it contains configuration parameters that are different from voice parameters.						
13	**LLDP-MED network policy**[2]	**127**	**8**	**0xfe08**	**0x0012bb**	**2**
ApplicationType: Video Conferencing (6),Policy: (Unknown(=1)/Defined(=0). Unknown, if phone is in booting stage or if switch doesn't support network policy TLV. Defined, if phone is operational stage and Networkpolicy TLV is received from the switch.), Tagged/Untagged, VlanId, L2 priority and DSCP. Note: Video Conferencing TLV is sent only from Video capable phones.						
14	**LLDP-MED location identification**[3]	**127**	**min len > 0, max len <= 511**	**-**	**0x0012bb**	**3**
ELIN data format: 10 digit emergency number configured on the switch. Civic Address: physical address data such as city, street number, and building information.						
15	**Extended power via MDI**	**127**	**7**	**0xfe07**	**0x0012bb**	**4**
PowerType -PD device PowerSource-PSE&local Power Priority -Unknown PowerValue.						
16	**LLDP-MED inventory hardware revision**	**127**	**min len > 0, max len <= 32**	**-**	**0x0012bb**	**5**
Hardware part number and revision						
17	**LLDP-MED inventory firmware revision**	**127**	**min len > 0, max len <= 32**	**-**	**0x0012bb**	**6**
BootROM revision						
18	**LLDP-MED inventory software revision**	**127**	**min len > 0, max len <= 32**	**-**	**0x0012bb**	**7**
Application (SIP) revision						
19	**LLDP-MED inventory serial number**	**127**	**min len > 0, max len <= 32**	**-**	**0x0012bb**	**8**
MAC Address (ASCII string)						

No	*Name*	*Type(7 bits) [0-6]*	*Length (9 bits) [7-15]*	*Type Length*	*Org. Unique Code (3 bytes)*	*Sub Type*
20	**LLDP-MED inventory manufacturer name**	**127**	**11**	**0xfe0b**	**0x0012bb**	**9**
Polycom						
21	**LLDP-MED inventory model name**	**127**	**min len > 0, max len <= 32**	**-**	**0x0012bb**	**10**
22	**LLDP-MED inventory asset ID**	**127**	**4**	**0xfe08**	**0x0012bb**	**11**
Empty (Zero length string)						
23	**End of LLDP DU**	**0**	**0**	**0x0000**	**-**	**-**

1 For other subtypes, refer to IEEE 802.1AB, March 2005.

2 For other application types, refer to TIA Standards 1057, April 2006.

3 At this time, this TLV is not sent by the phone.

PMD Advertise and Operational MAU

The following table lists values for the PMD Advertise and Operational MAU.

PMD Advertise and Operation MAU Type

Mode/Speed	*PMD Advertise Capability Bit*	*Operational MAU Type*
10BASE-T half duplex mode	1	10
10BASE-T full duplex mode	2	11
100BASE-T half duplex mode	4	15
100BASE-T full duplex mode	5	16
1000BASE-T half duplex mode	14	29
1000BASE-T full duplex mode	15	30
Unknown	0	0

Note: Default PMD Advertise Capability Values

By default, all phones have the PMD Advertise Capability set for 10HD, 10FD, 100HD, and 100FD bits.

Configuration Parameters

This section is a reference guide to the UC Software configuration parameters used to configure all phone features and functions. This section is useful if you want to read a detailed description of a particular configuration parameter or you would like to see the default or permitted values for that parameter. If you want to configure a specific feature, see the following sections:

- Set Up Basic Phone Features
- Set Up Advanced Phone Features
- Set Up Phone Audio Features
- Set Up User and Phone Security Features

The following parameters are included in this section:

- <apps/>
- <bg/>
- <button/>
- <call/>
- <callLists/>
- <device/>
- <dialplan/>
- <dir>
  - <broadsoft/>
  - <local/>
  - <corp/>
- <divert/>
- <dns/>
  - DNS-A
  - DNS-NAPTR
  - DNS-SRV
- <efk/>
- <exchange/>
- <feature/>
- <httpd/>
- <lcl/>
  -
  - <ml/>
  - <datetime/>
- <log/>
  - <level/> <change/>and<render/>

    - <sched/>
- <msg/>
- <mwi/>
- <nat/>
- <phoneLock/>
- <powerSaving/>
- <pres/>
- <prov/>
- <qos/>
- <reg/>
- <request/>
- <roaming_buddies/>
- <roaming_privacy/>
- <saf/>
- <se/>
    - <pat/>
    - <rt/>
- <sec/>
    - <encryption/>
    - <pwd/><length/>
    - <srtp/>
    - <dot1x><eapollogoff/>
    - <hostmovedetect/>
    - <TLS/>
        - <profile/>
        - <profileSelection/>
- <softkey/>
- <tcpIpApp/>
    - <dhcp/>
    - <dns/>
    - <ice/>
    - <sntp/>
    - <port/><rtp/>
    - <keepalive/>
    - <fileTransfer/>
- <tones/>
    - <DTMF/>

    - <chord/>
- <up/>
- <upgrade/>
- <video/>
    - <camera/>
    - <codecs/>
        - <codecPref/>
        - <profile/>
- <voice/>
    - <codecPref/>
    - <volume/>
    - <vad/>
    - <quality monitoring/>
    - <rxQoS/>
- <voIpProt/>
    - <server/>
    - <SDP/>
    - <SIP/>
- <webutility/>
    - <xmpp/>

<apps/>

The table Application Parameters lists <apps/> parameters you can use to control telephone notification events, state polling events, and push server controls. For more information, see the *Polycom Web Application Developer's Guide*.

Application Parameters

Parameter	*Permitted Values*	*Default*
apps.push.alertSound	**0 or 1**	**0**
If 0, there is no sound when an alert is pushed. If 1, there is sound.		
apps.push.messageType	**0 to 5**	**0**
Choose a priority level for push messages from the application server to the phone. **0**: (None) Discard push messages **1**: (Normal) Allows only normal push messages **2**: (Important) Allows only important push messages **3**: (High) Allows only priority push messages **4**: (Critical) Allows only critical push **5**: (All) Allows all push messages		

Parameter	Permitted Values	Default
apps.push.password	**string**	**null**
The password to access the push server URL.		
apps.push.secureTunnelEnabled	**0 or 1**	**1**
If 0, the Web server is not connected through a secure tunnel. If 1, the Web server is connected through a secure tunnel.		
apps.push.secureTunnelPort	**1 to 65535**	**443**
The port that the phone should use to communicate to the Web server when the secure tunnel is used.		
apps.push.secureTunnelRequired	**0 or 1**	**0**
If 0, communications to the Web server do not require a secure tunnel. If 1, communications require a secure tunnel.		
apps.push.serverRootURL	**URL**	**null**
The URL of the application server you enter here is combined with the phone address and sent to the phone's browser. For example, if the application server root URL is http://172.24.128.85:8080/sampleapps and the relative URL is /examples/sample.html, the URL that is sent to the microbrowser is http://172.24.128.85:8080/sampleapps/examples/sample.html. Can be either HTTP or HTTPS.		
apps.push.username	**string**	**null**
The user name to access the push server URL. Note: To enable the push functionality, the parameters `apps.push.username` and `apps.push.password` must be set (not null).		
apps.statePolling.password	**string**	**null**
Enter the password that the phone requires to authenticate phone state polling.		
apps.statePolling.URL	**URL**	**null**
The URL to which the phone sends call processing state/device/network information. The protocol used can be either HTTP or HTTPS. Note: To enable state polling, the parameters `apps.statePolling.URL`, `apps.statePolling.username`, and `apps.statePolling.password` must be set to non-null values.		
apps.statePoling.responseMode	**0 or 1**	**1**
The mode of sending requested polled data. If 1, requested polled data is sent to a configured URL. If 0, the data is sent in the HTTP response.		
apps.statePolling.username	**string**	**null**
Enter the user name that the phone requires to authenticate phone state polling.		
apps.telNotification.callStateChangeEvent	**0 or 1**	**0**
If 0, call state change notification is disabled. If 1, notification is enabled.		
apps.telNotification.incomingEvent	**0 or 1**	**0**
If 0, incoming call notification is disabled. If 1, notification is enabled.		
apps.telNotification.lineRegistrationEvent	**0 or 1**	**0**
If 0, line registration notification is disabled. If 1, notification is enabled.		

Parameter	*Permitted Values*	*Default*
apps.telNotification.networkUpEvent	**0 or 1**	**0**
If 0, network up notification is disabled. If 1, notification is enabled.		
apps.telNotification.offhookEvent	**0 or 1**	**0**
If 0, off-hook notification is disabled. If 1, notification is enabled.		
apps.telNotification.onhookEvent	**0 or 1**	**0**
If 0, on-hook notification is disabled. If 1, notification is enabled.		
apps.telNotification.outgoingEvent	**0 or 1**	**0**
If 0, outgoing call notification is disabled. If 1, notification is enabled.		
apps.telNotification.uiInitializationEvent	**0 or 1**	**0**
If 0, user interface initialization notification is disabled. If 1, notification is enabled.		
apps.telNotification.URL	**URL**	**null**
The URL to which the phone sends notifications of specified events. Can be either HTTP or HTTPS.		
apps.telNotification.x.URL	**URL**	**null**
The URL to which the phone sends notifications of specified events, where x 1 to 9. Can be either HTTP or HTTPS.		
apps.telNotification.userLogInOutEvent	**0 or 1**	**0**
If 0, user login/logout notification is disabled. If 1, notification is enabled.		
apps.ucdesktop.adminEnabled[1]	**0 or 1**	**1**
If 0, the Polycom Desktop Connector is disabled on the administrative level. If 1, it is enabled on the administrative level.		
apps.ucdesktop.desktopUserName	**string**	**null**
The user's name, supplied from the user's computer. For example, bsmith.		
apps.ucdesktop.enabled	**0 or 1**	**0**
If 0, the Polycom Desktop Connector is disabled for users. If 1, it is enabled for users.		
apps.ucdesktop.ServerAddress	**string**	**null**
The user's computer as a fully qualified domain name (FQDN). For example, computer@yourcompany.com.		
apps.ucdesktop.ServerPort	**1 to 65535**	**24800**
The port number. Note: This value should be the same as the one that is used on the user's computer, otherwise the connection is not established.		

[1] Change causes phone to restart or reboot.

<bg/>

The parameters listed in the table Background Parameters define the backgrounds you can display on the CX5500 system.

Background Parameters

Parameter	*Permitted Values*	*Default*
bg.color.selection	**w,x**	**1,1**
Set the background. Specify which type of background (w) and index (x) for that type is selected on reboot. The default selection is 2,1 the first solid background. Use w=1 and x=1 (1,1) to select the built-in image. Use w=2 and x= 1 to 6 to select one of the six background `bm` images.		
bg.color.bm.x.name **Phone screen background image file**	**URL or file path of a BMP or PNG image**	
The name of the image file (including extension). The six (x: 1 to 6) default screen background images are: x=1: `Leaf.png` x=2: `Sailboat.png` x=3: `Beach.png` x=4: `Palm.png` x=5 `Jellyfish.png` x=6 `Mountain.png` Note: If the file is missing or unavailable, the built-in default solid pattern is displayed.		

<button/>

You can configure the color of line keys and soft keys using the <button/> parameter using the parameters in the table Soft Key Button Parameters.

Soft Key Button Parameters

Parameter	*Permitted Values*	*Default*
button.color.selection.x.y.modify	**any string**	
The label color for soft keys and line key labels associated with the defined colored backgrounds. These values can be modified locally by the user. The format is: `rgbHILo, <parameter list>`. For example: `rbgHiLo, 51, 255, 68, 255, 0, 119` is the default button color associated with the built-in background.		
button.gray.selection.x.y.modify	**any string**	
The label color for soft keys and line key labels associated with the defined gray backgrounds. These values can be modified locally by the user. The format is: `rgbHILo, <parameter list>`. By default, all defaults are set to `none`.		

<call/>

The phone supports an optional per-registration feature that enables automatic call placement when the phone is off-hook.

The phone supports a per-registration configuration that determines which events will cause the missed-calls counter to increment.

You can enable/disable missed call tracking on a per-line basis.

The table Call Parameters defines per-site and per-phone configuration parameters. In the following table, x is the registration number. For the CX5500 system, x=1-16.

Call Parameters

Parameter	*Permitted Values*	*Default*
call.advancedMissedCalls.addToReceivedList	**0 or 1**	**0**
Applies to calls on that are answered remotely. If 0, calls answered from the remote phone are not added to the local receive call list. If 1, calls answered from the remote phone are added to the local receive call list.		
call.advancedMissedCalls.enabled	**0 or 1**	**1**
If 1, improved missed call handling for shared lines is enabled (shared lines can correctly count missed calls). If 0, the old missed call handling is used for shared lines (shared lines may not correctly count missed calls).		
call.advancedMissedCalls.reasonCodes	**comma-separated list of indexes**	**200**
A comma separated list of reason code indexes that are interpreted to mean that a call should not be considered as a missed call.		
call.autoAnswer.micMute	**0 or 1**	**1**
If 0, the microphone is active immediately after a call is auto-answered. If 1, the microphone is initially muted after a call is auto-answered.		
call.autoAnswer.ringClass	**see the list of ring classes in <rt/>**	**ringAutoAnswer**
The ring class to use when a call is to be automatically answered using the auto-answer feature. If set to a ring class with a type other than `answer` or `ring-answer`, the setting will be overridden such that a ringtone of `visual` (no ringer) applies.		
call.autoOffHook.x.enabled[1] **Enable or disable the feature**	**0 or 1**	**0**
call.autoOffHook.x.contact[1] **The contact address to where the call is placed**	**a SIP URL**	**Null**
call.autoOffHook.x.protocol[1] **The calling protocol to use**	**SIP**	**Null**
If `enabled` is set to 0, no call is placed automatically when the phone goes off hook, and the other parameters are ignored. If enabled is set to 1, a call is automatically placed to the `contact` using the calling `protocol`, when the phone goes off hook. The `contact` must be an ASCII-encoded string containing digits, either the user part of a SIP URL (for example, 6416), or a full SIP URL (for example, 6416@polycom.com).		

Parameter	*Permitted Values*	*Default*
call.callsPerLineKey	**1-4, 1-8, 1-24**	**4, 8, 24**
Set the maximum number of concurrent calls per line key. This parameter applies to all registered lines. The permitted range is 1 to 8 and the default is 8. Note that this parameter may be overridden by the per-registration parameter of `reg.x.callsPerLineKey`.		
call.callWaiting.enable	**0 or 1**	**1**
If 1, the phone alerts you to an incoming call while you are in an active call. If 0, you are not alerted to incoming calls while in an active call and the incoming call is treated as if you did not answer it. If 1, and you end the active call during a second incoming call, you are alerted to the second incoming call.		
call.callWaiting.ring[1]	**beep, ring, silent**	**beep**
Specifies the ringtone of incoming calls when another call is active. If set to Null, the default value is beep.		
call.dialtoneTimeOut[1]	**positive integer**	**60**
The time is seconds that a dial tone will play before a call is dropped. If set to 0, the call is not dropped.		
call.directedCallPickupString[1]	**star code**	***97**
The star code to initiate a directed call pickup. Note: The default value supports the BroadWorks calls server only. You must change the value if your organization uses a different call server.		
call.donotdisturb.perReg[1]	**0 or 1**	**0**
This parameter determines if the Do-Not-Disturb feature will apply to all registrations on the phone (globally), or apply on a per-registration basis. If 0, DND will apply to all registrations on the phone when it is active. If 1, the user can activate DND on a per-registration basis. Note: If `voIpProt.SIP.serverFeatureControl.dnd` is set to 1 (enabled), this parameter is ignored.		
call.enableOnNotRegistered[1]	**0 or 1**	**1**
If 1, users can make calls when the phone is not registered. If 0, calls are not permitted without registration.		
call.hold.localReminder.enabled[1]	**0 or 1**	**0**
If 1, users are reminded of calls that have been on hold for an extended period of time. If 0, there is no hold reminder.		
call.hold.localReminder.period[1]	**non-negative integer**	**60**
Specify the time in seconds between subsequent hold reminders.		
call.hold.localReminder.startDelay[1]	**non-negative integer**	**90**
Specify a time in seconds to wait before the initial hold reminder.		
call.internationalDialing.enabled	**0 or 1**	**1**
Use this parameter to enable or disable the key tap timer that converts a double tap of the asterisk “*” symbol to the “+” symbol used to indicate an international call. By default, this parameter is enabled so that a quick double tap of “*” converts immediately to “+”. To enter a double asterisk “**”, tap “*” once and wait for the key tap timer to expire to enter a second “*”. When you disable this parameter, you cannot dial”+” and you must enter the international exit code of the country you are calling from to make international calls. Changes you make to this parameter cause a restart or reboot. Note that this parameter applies to all numeric dial pads on the phone, including for example, the contact directory.		

Parameter	*Permitted Values*	*Default*
call.lastCallReturnString[1]	**string of maximum length 32**	***69**
The string sent to the server when the user selects the last call return action. The string is usually a star code.		
call.localConferenceCallHold[1]	**0 or 1**	**0**
If set to 0, a hold will happen for all legs when conference is put on hold. If set to 1, only the host is out of the conference, all other parties in conference continue to talk.		
call.localConferenceEnabled[1]	**0 or 1**	**0**
If set to 0, the Conference and Join soft keys do not display during an active call and you cannot establish conferences on the phone. If set to 1, the Conference and Join soft keys display during an active call and you can establish conferences on the phone.		
call.missedCallTracking.x.enabled[1]	**0 or 1**	**1**
If set to 1, missed call tracking is enabled. If `call.missedCallTracking.x.enabled` is set to 0, then missedCall counter is not updated regardless of what `call.serverMissedCalls.x.enabled` is set to (and regardless of how the server is configured). There is no Missed Call List provided under **Settings > Features** of the phone. If `call.missedCallTracking.x.enabled` is set to 1 and call.serverMissedCalls.x.enabled is set to 0, then the number of missedCall counter is incremented regardless of how the server is configured. If `call.missedCallTracking.x.enabled` is set to 1 and `call.serverMissedCalls.x.enabled` is set to 1, then the handling of missedCalls depends on how the server is configured.		
call.offeringTimeOut[1]	**positive integer**	**60**
Specify a time in seconds that an incoming call will ring before the call is dropped, 0=infinite. Note**:** The call diversion, no answer feature will take precedence over this feature if enabled.		
call.parkedCallRetrieveString[1]	**star code**	**Null**
The star code used to initiate retrieval of a parked call.		
call.rejectBusyOnDnd[1]	**0 or 1**	**1**
If 1, and DND is turned on, the phone rejects incoming calls with a busy signal. If set to 0, and DND is turned on, the phone gives a visual alert of incoming calls and no audio ringtone alert. Note: This parameter does not apply to shared lines since not all users may want DND enabled.		
call.ringBackTimeOut[1]	**positive integer**	**60**
Specify a time in seconds to allow an outgoing call to remain in the ringback state before dropping the call, 0=infinite.		
call.serverMissedCall.x.enabled[1]	**0 or 1**	**0**
If 0, all missed-call events will increment the counter. If set to 1, only missed-call events sent by the server will increment the counter. Note: This feature is supported with the BroadSoft® Synergy call server only (previously known as Sylantro).		
call.shared.disableDivert[1]	**0 or 1**	**1**
If set to 1, the diversion feature for shared lines is disabled. Note: This feature is disabled on most call servers.		

Parameter	*Permitted Values*	*Default*
call.shared.exposeAutoHolds[1]	**0 or 1**	**0**
If 1, a re-INVITE will be sent to the server when setting up a conference on a shared line. If 0, no re-INVITE will be sent to the server.		
call.shared.oneTouchResume[1]	**0 or 1**	**0**
If set to 1, all users on a shared line can resume held calls by pressing the shared line key. If more than one call is on hold, the first held call is selected and resumed. If set to 0, selecting the shared line opens all current calls that the user can choose from.		
call.shared.seizeFailReorder[1]	**0 or 1**	**1**
If set to 1, play re-order tone locally on shared line seize failure.		
call.singleKeyPressConference[1]	**0 or 1**	**0**
If set to 1, the conference will be setup after a user presses the **Conference** soft key or **Conference** key the first time. Also, all sound effects (dial tone, DTMF tone while dialing and ringing back) are heard by all existing participants in the conference. If set to 0, sound effects are only heard by conference initiator (original behavior).		
call.stickyAutoLineSeize[1]	**0 or 1**	**0**
If set to 1, the phone uses sticky line seize behavior. This will help with features that need a second call object to work with. The phone will attempt to initiate a new outgoing call on the same SIP line that is currently in focus on the LCD (this was the behavior in SIP 1.6.5). Dialing through the call list when there is no active call will use the line index for the previous call. Dialing through the call list when there is an active call will use the current active call line index. Dialing through the contact directory will use the current active call line index. If set to 0, the feature is disabled (this was the behavior in SIP 1.6.6). Dialing through the call list will use the line index for the previous call. Dialing through the contact directory will use a random line index. Note: This may fail due to glare issues in which case the phone may select a different available line for the call.		
call.stickyAutoLineSeize.onHookDialing[1]	**0 or 1**	**0**
If `call.stickyAutoLineSeize` is set to 1, this parameter has no effect. The regular stickyAutoLineSeize behavior is followed. If `call.stickyAutoLineSeize` is set to 0 and this parameter is set to 1, this overrides the stickyAutoLineSeize behavior for hot dial only. (Any new call scenario seizes the next available line.) If `call.stickyAutoLineSeize` is set to 0 and this parameter is set to 0, there is no difference between hot dial and new call scenarios. Note: A hot dial occurs on the line which is currently in the call appearance. Any new call scenario seizes the next available line.		
call.transferOnConferenceEnd[1]	**0 or 1**	**1**
The behavior when the conference host exits a conference. If 0, all parties are disconnected when the conference host exits the conference. If 1, the other parties are left connected when the host exits the conference (the host performs an attended transfer to the other parties).		
call.urlModeDialing[1]	**0 or 1**	**0**
If 0, URL dialing is disabled. If 1, URL dialing is enabled.		

[1] Change causes phone to restart or reboot.

<callLists/>

The call lists (or call log) parameters are listed in the table Call List (Call Log) Parameters.

Call List (Call Log) Parameters

Parameter	*Permitted Values*	*Default*
callLists.collapseDuplicates	**0 or 1**	**1**
If 0, all calls are archived and presented in the call lists. If 1, consecutive incomplete between the same party in the same direction (outgoing/incoming) are collapsed into one record with the most recent call displaying.		
callLists.logConsulationCalls	**0 or 1**	**0**
If 1, all consultation calls are logged. (Calls made to a third party—while the original party is on hold—when settings up a conference call are called consultation calls.) If 0, consultation calls are not logged.		
callLists.size	**10 to 99**	**99**
The maximum number of retained records of each type (incoming, outgoing, and missed). When the maximum number is reached, new records will overwrite existing records. You can clear the list using the phone's menu system. If you want to prevent the records from uploading to the provisioning server, enter a false URL in the CALL_LISTS_DIRECTORY field in the master configuration file.		
callLists.writeDelay.journal	**1 to 600**	**5**
The delay (in seconds) before changes due to an in-progress call are flushed to the file system as a journal.		
callLists.writeDelay.terminated	**10 to 600**	**60**
The minimum period between writing out the complete XML file to the local file system and, optionally, to the provisioning server.		

<device/>

The <device/> parameters—also known as device settings—contain default values that you can use to configure basic settings for multiple phones.

Web Info: Default Device Parameter Values

The default values for the `<device/>` parameters are set at the factory when the phones are shipped. For a list of the default values, see the latest Shipping Configuration Notice.

Polycom provides a global `device.set` parameter that you can enable for software installation and changes to device parameters. Once you have completed the software installation or made configuration changes to device parameters, remove `device.set`. Disabling the parameter after the initial software installation prevents the phones from rebooting and triggering a reset of device parameters that users may have changed after the initial installation.

Each `<device/>` parameter has a corresponding `.set` parameter that enables or disables the value for that device parameter. Enable the corresponding `.set` parameter for each parameter you want to apply.

Settings: Each <device/> Parameter has a Corresponding .set Parameter with One Exception

Note that each `<device/>` parameter has a corresponding `.set` parameter that enables or disables the parameter. There is one exception to this rule: the `device.sec.TLS.customDeviceCertX.set` parameter applies to both `device.sec.TLS.customDeviceCertX.publicCert` and to `device.sec.TLS.customDeviceCertX.privateKey`.

Use Caution When Changing Device Parameters

Use caution when changing `<device/>` parameters as incorrect settings may apply the same IP address to multiple phones.

Note that some parameters may be ignored. For example, if DHCP is enabled it will still override the value set with `device.net.ipAddress`.

Though individual parameters are checked to see whether they are in range, the interaction between parameters is not checked. If a parameter is out of range, an error message will display in the log file and parameter will not be used.

Incorrect configuration can put the phones into a reboot loop. For example, server A has a configuration file that specifies that server B should be used, and server B has a configuration file that specifies that server A should be used.

To detect errors, including IP addess conflicts, Polycom recommends that you test the new configuration files on two phones before initializing all phones.

The table Device Parameter Types outlines the three types of <device/> parameters, their permitted values, and the default value.

Device Parameter Types

Parameter	*Permitted Values*	*Default*
device.set[1]	**0 or 1**	**0**
If set to 0, do not use any `device.xxx` fields to set any parameters. Set this to 0 after the initial software installation. If set to 1, use the `device.xxx` fields that have `device.xxx.set=1`. Set this to 1 only for the initial software installation.		
device.xxx[1]	**string**	**Null**
Configuration parameter.		
device.xxx.set[1]	**0 or 1**	**0**
If set to 0, do not use the `device.xxx` value. If set to 1, use the `device.xxx` value. For example, if `device.net.ipAddress.set=1`, then use the value set for `device.net.ipAddress`.		

Parameter	*Permitted Values*	*Default*

[1] Change causes phone to restart or reboot

The table Device Parameters lists each of the <device/> parameters that you can configure.

Device Parameters

Parameter	*Permitted Values*	*Default*
device.auth.localAdminPassword	**string (32 character max)**	**Null**
The phone’s local administrative password. The minimum length is defined by sec.pwd.length.admin.		
device.auth.localUserPassword	**string (32 character max)**	**Null**
The phone user’s local password. The minimum length is defined by sec.pwd.length.user.		
device.baseProfile	**Generic, Lync**	**Null**
Choose the Base Profile that the phone will operate with.		
device.cma.mode	**Static, Auto, Disabled**	**Null**
The mode the phone uses to retrieve the Polycom CMA server IP address. `Auto` The phone uses SRV lookup. `Disabled` The phone does not contact the server. `Static` The phone uses the server name or IP address specified in `device.cma.serverName`. Note: If you will modify this parameter, the phone will re-provision. The phone may also reboot if the configuration on the CMA server has changed.		
device.cma.serverName	**server name or IP address**	**Null**
Polycom CMA server name or IP address. Note: If you will modify this parameter, the phone will re-provision. The phone may also reboot if the configuration on the CMA server has changed.		
device.dhcp.bootSrvOpt[1]	**Null, 128 to 254**	**Null**
When the boot server is set to Custom or Custom+Option66, specify the numeric DHCP option that the phone will look for.		
device.dhcp.bootSrvOptType[1]	**IP or String**	**Null**
The type of DHCP option in which the phone will look for its provisioning server (if `device.dhcp.bootSrvUseOpt` is set to `Custom`). If IP, the IP address provided must specify the format of the provisioning server. If String, the string provided must match one of the formats specified by `device.prov.serverName`.		

Parameter	*Permitted Values*	*Default*
device.dhcp.bootSrvUseOpt[1]	**Default, Custom, Static, CustomAndDefault**	**Null**
`Default` The phone will look for option number 66 (string type) in the response received from the DHCP server. The DHCP server should send address information in option 66 that matches one of the formats described for `device.prov.serverName`. `Custom` The phone will look for the option number specified by `device.dhcp.bootSrvOpt`, and the type specified by `device.dhcp.bootSrvOptType` in the response received from the DHCP server. `Static` The phone will use the boot server configured through the provisioning server `device.prov.*` parameters. `Custom and Default` The phone will use the custom option first or use Option 66 if the custom option is not present.		
device.dhcp.enabled[1]	**0 or 1**	**Null**
If 0, DHCP is disabled. If 1, DHCP is enabled.		
device.dhcp.option60Type[1]	**Binary, ASCII**	**Null**
The DHCP option 60 type. `Binary`: vendor-identifying information is in the format defined in RFC 3925. `ASCII`: vendor-identifying information is in ASCII format.		
device.dhcp.dhcpVlanDiscUseOpt[1]	**Disabled, Fixed, Custom**	**Null**
VLAN Discovery. `Disabled`, no VLAN discovery through DHCP. `Fixed`, use predefined DHCP vendor-specific option values of 128, 144, 157 and 191 (`device.dhcp.dhcpVlanDiscOpt` will be ignored). `Custom`, use the number specified by `device.dhcp.dhcpVlanDiscOpt`.		
device.dhcp.dhcpVlanDiscOpt[1]	**128 to 254**	**Null**
The DHCP private option to use when `device.dhcp.dhcpVlanDiscUseOpt` is set to `Custom`.		
device.dns.altSrvAddress[1]	**server address**	**Null**
The secondary server to which the phone directs Domain Name System (DNS) queries.		
device.dns.domain[1]	**string**	**Null**
The phone's DNS domain.		
device.dns.serverAddress[1]	**string**	**Null**
The primary server to which the phone directs Domain Name System queries.		
device.host.hostname[1]	**string**	**Null**
This parameter enables you to specify a hostname for the phone when using DHCP by adding a hostname string to the phone's configuration. If `device.host.hostname.set=1`, and `device.host.hostname=Null`, the DHCP client uses Option 12 to send a predefined hostname to the DHCP registration server using `Polycom_<MACaddress>`. Note that the maximum length of the hostname string is <=255 bytes. The valid character set is defined in RFC1035.		
device.logincred.domain[1]	**string**	**Null**
The CMA account domain. Note: If you will modify this parameter, the phone will re-provision. The phone may also reboot if the configuration on the CMA server has changed.		

Parameter	*Permitted Values*	*Default*
device.logincred.password[1]	**string**	**Null**
The CMA account password. Note: If you will modify this parameter, the phone will re-provision. The phone may also reboot if the configuration on the CMA server has changed.		
device.logincred.user[1]	**string**	**Null**
The CMA account user name. Note: If you will modify this parameter, the phone will re-provision. The phone may also reboot if the configuration on the CMA server has changed.		
device.net.cdpEnabled[1]	**0 or 1**	**Null**
If set to 1, the phone will attempt to determine its VLAN ID and negotiate power through CDP.		
device.net.dot1x.anonid[1]	**string**	**Null**
EAP-TTLS and EAP-FAST only. The anonymous identity (user name) for 802.1X authentication.		
device.net.dot1x.eapFastInBandProv[1]	**0 or 1**	**Null**
EAP-FAST only, optional. Choose 1 to enable EAP In-Band Provisioning by server unauthenticated PAC provisioning using anonymous Diffie-Hellman key exchange. Choose 0 to disable EAP In-Band Provisioning. Reserved for Future Use—Choose 2 to enable EAP In-band provisioning by server authenticated PAC provisioning using certificate based server authentication.		
device.net.dot1x.enabled[1]	**0 or 1**	**Null**
If 0, 802.1X authentication is disabled. If 1, 802.1X authentication is enabled.		
device.net.dot1x.identity[1]	**string**	**Null**
The identity (user name) for 802.1X authentication.		
device.net.dot1x.method	**EAP-None, EAP-TLS, EAP-PEAPv0-MSCHAPv2, EAP-PEAPv0-GTC, EAP-TTLS-MSCHAPv2, EAP-TTLS-GTC, EAP-FAST, EAP-MD5**	**Null**
Specify the 802.1X authentication method, where `EAP-NONE` means no authentication.		
device.net.dot1x.password[1]	**string**	**Null**
The password for 802.1X authentication. This parameter is required for all methods except EAP-TLS.		
device.net.ether1000BTClockLAN[1]	**Auto, Slave, Master**	**Null**
The mode of the LAN clock. Polycom recommends that you do not change this value unless you have Ethernet connectivity issues.		
device.net.ether1000BTClockPC[1]	**Auto, Slave, Master**	**Null**
The mode of the PC clock. Polycom recommends that you do not change this value unless you have Ethernet connectivity issues.		

Parameter	*Permitted Values*	*Default*
device.net.etherModeLAN[1]	**Auto, 10HD, 10FD, 100HD, 100FD, 100FD**	**Null**
The LAN port mode that sets the network speed over Ethernet. HD means half-duplex and FD means full duplex. Note: Polycom recommends that you do not change this setting.		
device.net.etherModePC[1]	**Disabled, Auto, 10HD, 10FD, 100HD, 100FD, 100FD**	**Auto**
The PC port mode that sets the network speed over Ethernet. If set to `Disabled`, the PC port is disabled. HD means half duplex and FD means full duplex.		
device.net.etherStormFilter[1]	**0 or 1**	**Null**
If 1, DoS Storm Prevention is enabled and received Ethernet packets are filtered to prevent TCP/IP stack overflow caused by bad data or too much data. If 0, DoS Storm Prevention is disabled.		
device.net.ipAddress[1]	**string**	**Null**
The phone's IP address. Note: This parameter is disabled when DHCP is enabled (`device.dhcp.enabled` is set to 1.		
device.net.IPgateway[1]	**dotted-decimal IP address**	**Null**
The phone's default router.		
device.net.lldpEnabled[1]	**0 or 1**	**Null**
If set to 1, the phone will attempt to determine its VLAN ID and negotiate power through LLDP.		
device.net.subnetMask[1]	**dotted-decimal subnet mask**	**Null**
The phone's subnet mask. Note: This parameter is disabled when DHCP is enabled (`device.dhcp.enabled` is set to 1).		
device.net.vlanId[1]	**Null, 0-4094**	**Null**
The phone's 802.1Q VLAN identifier. If Null, no VLAN tagging.		
device.pacfile.data[1]	**String**	**Null**
EAP-FAST only, optional. The PAC file (base 64 encoded). To generate a base 64-encoded PAC file, generate the PAC file using your authentication server and then convert it to base 64. You can convert the file to base 64 using the following openssl commands: `$ openssl enc -base64 -in myfile -out myfile.b64`		
device.pacfile.password[1]	**String**	**Null**
EAP-FAST only, optional. The password for the PAC file.		
device.prov.maxRedunServers[1]	**1 to 8**	**Null**
The maximum number of IP addresses that will be used from the DNS.		

Parameter	*Permitted Values*	*Default*
device.prov.password[1]	**string**	**Null**
The password for the phone to log in to the provisioning server. Note that a password may not be required. Note: If you modify this parameter, the phone will re-provision. The phone may also reboot if the configuration on the provisioning server has changed.		
device.prov.redunAttemptLimit[1]	**1 to 10**	**Null**
The maximum number of attempts to attempt a file transfer before the transfer fails. When multiple IP addresses are provided by DNS, 1 attempt is considered to be a request sent to each server.		
device.prov.redunInterAttemptDelay[1]	**0 to 300**	**Null**
The number of seconds to wait after a file transfer fails before retrying the transfer. When multiple IP addresses are returned by DNS, this delay only occurs after each IP has been tried.		
device.prov.serverName	**dotted-decimal IP address, domain name string, or URL**	**Null**
The IP address, domain name, or URL of the provisioning server, followed by an optional directory and optional configuration filename. This parameter is used if DHCP is disabled (`device.dhcp.enabled` is `0`), if the DHCP server does not send a boot server option, or if the boot server option is static (`device.dhcp.bootSrvUseOpt` is `static`). Note: If you modify this parameter, the phone will re-provision. The phone may also reboot if the configuration on the provisioning server has changed.		
device.prov.serverType[1]	**FTP, TFTP, HTTP, HTTPS, FTPS**	**Null**
The protocol the phone uses to connect to the provisioning server. Note: Active FTP is not supported for BootROM version 3.0 or later. Note: Only implicit FTPS is supported.		
device.prov.upgradeServer	**string**	**Null**
The server used by the Polycom Web Configuration Utility's software upgrade feature. The server checks this URL for new software files.		
device.prov.tagSerialNo	**0 or 1**	**Null**
If 0, the phone's serial number (MAC address) is not included in the User-Agent header of HTTPS/HTTPS transfers and communications to the microbrowser and Web browser. If 1, the phone's serial number is included.		
device.prov.user	**string**	**Null**
The user name required for the phone to log in to the provisioning server (if required). Note: If you modify this parameter, the phone will re-provision. The phone may also reboot if the configuration on the provisioning server has changed.		
device.prov.ztpEnabled	**0 or 1**	**Null**
If 0, Disable the ZTP feature. If 1, enable the ZTP feature.		
device.sec.configEncryption.key[1]	**string**	**Null**
The configuration encryption key used to encrypt configuration files. For more information, see Encrypt Configuration Files.		

Parameter	*Permitted Values*	*Default*
device.sec.TLS.customCaCert1 (TLS Platform Profile 1) **device.sec.TLS.customCaCert2 (TLS Platform Profile 2)**	**string, PEM format**	**Null**
The custom certificate to use for TLS Platform Profile 1 and TLS Platform Profile 2 and TLS Application Profile 1 and TLS Application Profile 2 `device.sec.TLS.profile.caCertList` must be configured to use a custom certificate. Custom CA certificate cannot exceed 4096 bytes total size.		
device.sec.TLS.customDeviceCert1.publicCert **device.sec.TLS.customDeviceCert2.publicCert**	**Enter the signed custom device certificate in PEM format (X.509)**	**Null**
device.sec.TLS.customDeviceCert1.privateKey **device.sec.TLS.customDeviceCert2.privateKey**	**Enter the corresponding signed private key in PEM format (X.509)**	**Null**
device.sec.TLS.customDeviceCert1.set **device.sec.TLS.customDeviceCert2.set**	**0 or 1**	**0**
Note that you use a single `.set` parameter to enable or disable only these two related `<device/>` parameters - `device.sec.TLS.customDeviceCertX.publicCert` and `device.sec.TLS.customDeviceCertX.privateKey`. All other `<device/>` parameters have their own corresponding `.set` parameter that will enable or disable that parameter. Size constraints are: 4096 bytes for the private key, 8192 bytes for the device certificate.		
device.sec.TLS.profile.caCertList1 (TLS Platform Profile 1) **device.sec.TLS.profile.caCertList2 (TLS Platform Profile 2)**	**Builtin, BuiltinAndPlatform1, BuiltinAndPlatform2, All, Platform1, Platform2, Platform1AndPlatform2**	**Null**
Choose the CA certificate(s) to use for TLS Platform Profile 1 and TLS Platform Profile 2 authentication: The built-in default certificate The built-in and Custom #1 certificates The built-in and Custom #2 certificates Any certificate (built in, Custom #1 or Custom #2) Only the Custom #1 certificate Only the Custom #2 certificate Either the Custom #1 or Custom #2 certificate		
device.sec.TLS.profile.cipherSuite1 (TLS Platform Profile 1) **device.sec.TLS.profile.cipherSuite2 (TLS Platform Profile 2)**	**string**	**Null**
The cipher suites to use for TLS Platform Profile 1 and TLS Platform Profile 2)		
device.sec.TLS.profile.cipherSuiteDefault1 (TLS Platform Profile 1) **device.sec.TLS.profile.cipherSuiteDefault2 (TLS Platform Profile 2)**	**0 or 1**	**Null**
The cipher suite to use for TLS Platform Profile 1 and TLS Platform profile 2. If set to 0, the custom cipher suite will be used. If set to 1, the default cipher suite will be used.		
device.sec.TLS.profile.deviceCert1 (TLS Platform Profile 1) **device.sec.TLS.profile.deviceCert2 (TLS Platform Profile 2)**	**Builtin, Platform1, Platform2**	**Null**
Choose the device certificate(s) for TLS Platform Profile 1 and TLS Platform Profile 2 to use for authentication.		

Parameter	*Permitted Values*	*Default*
device.sec.TLS.profile.profileSelection.dot1x	**PlatformProfile1, PlatformProfile2**	**Null**
Choose the TLS Platform Profile to use for 802.1X, either TLS Platform Profile 1 or TLS Platform Profile 2.		
device.sec.TLS.profileSelection.provisioning[1]	**PlatformProfile1, PlatformProfile2**	**Null**
The TLS Platform Profile to use for provisioning, either TLS Platform Profile 1 or TLS Platform Profile 2.		
device.sec.TLS.profileSelection.syslog[1]	**PlatformProfile1, PlatformProfile2**	**Null**
The TLS Platform Profile to use for syslog, either TLS Platform Profile 1 or TLS Platform Profile 2.		
device.sec.TLS.prov.strictCertCommonNameValidation	**0 or 1**	**1**
If set to 1, provisioning always verifies the server certificate for commonName/SubjectAltName match with the server hostname that the phone is trying to connect.		
device.sec.TLS.syslog.strictCertCommonNameValidation	**0 or 1**	**1**
If set to 1, syslog always verifies the server certificate for commonName/SubjectAltName match with the server hostname that the phone is trying to connect.		
device.sntp.gmtOffset	**-43200 to 46800**	**Null**
The GMT offset—in seconds—to use for daylight savings time, corresponding to -12 to +13 hours.		
device.sntp.serverName	**dotted-decimal IP address or domain name string**	**Null**
The SNTP server from which the phone will obtain the current time.		
device.syslog.facility	**0 to 23**	**Null**
A description of what generated the log message. For more information, see section 4.1.1 or RFC 3164.		
device.syslog.prependMac[1]	**0 or 1**	**Null**
If 1, the phone's MAC address is pre-pended to the log message sent to the syslog server.		
device.syslog.renderLevel[1]	**0 to 6**	**Null**
Specify the logging level that will display in the syslog. Note that when you choose a log level, you are including all events of an equal or greater severity level and excluding events of a lower severity level. The logging level you choose determines the lowest severity of events that will be logged. **0** or **1**: SeverityDebug(7). **2** or **3**: SeverityInformational(6). **4**: SeverityError(3). **5:** SeverityCritical(2). **6:** SeverityEmergency(0).		
device.syslog.serverName	**dotted-decimal IP address OR domain name string**	**Null**
The syslog server IP address or domain name string.		
device.syslog.transport	**None, UDP, TCP, TLS**	**Null**
The transport protocol that the phone will use to write to the syslog server. If set to None, transmission is turned off but the server address is preserved.		

Parameter	*Permitted Values*	*Default*

[1] Change causes phone to restart or reboot.

<dialplan/>

The parameters listed in the table Dial Plan (Digit Map) Parameters enable you to create a specific routing path for outgoing SIP calls independent of other *default* configurations.

The dial plan (or digit map) is not applied against Placed Call List, Voicemail, last call return, remote control dialed numbers, or on-hook dialing.

Dial Plan (Digit Map) Parameters

Parameter	*Permitted Values*	*Default*
dialplan.applyToCallListDial[1]	**0 or 1**	**1**
If 0, the dial plan does not apply to numbers dialed from the Received Call List or Missed Call List. If 1, the dial play is applied to numbers dialed from the received call and missed call lists, including sub-menus.		
dialplan.applyToDirectoryDial[1]	**0 or 1**	**0**
If 0, the dial plan is not applied to numbers dialed from the directory or speed dial list. If 1, the dial plan is applied to numbers dialed from the directory or speed dial, including auto-call contact numbers.		
dialplan.applyToForward[1]		
If 0, the dial plan does not apply to forwarded calls. If 1, the dial plan applies to forwarded calls.		
dialplan.applyToTelUriDial[1]	**0 or 1**	**1**
If 0, the dial plan does not apply to URI dialing. If 1, the dial plan applies to URI dialing.		
dialplan.applyToUserDial[1]	**0 or 1**	**1**
If 0, the dial plan does not apply to calls made when the user presses the **Dial** soft key to place a call. If 1, the dial plan applies to calls placed using the **Dial** soft key.		
dialplan.applyToUserSend[1]	**0 or 1**	**1**
If 0, the dial plan does not apply to calls placed when the user presses the **Send** soft key to place a call. If 1, the dial plan applies to calls placed using the **Send** soft key.		
dialplan.digitmap[1]	**string compatible with the digit map feature of MGCP described in 2.1.5 of RFC 3435**	**[2-9]11|0T| +011xxx.T| 0[2-9]xxxxxxxxx| +1[2-9]xxxxxxxx| [2-9]xxxxxxxxx| [2-9]xxxT**
The digit map used for the dial plan. The string is limited to 2560 bytes and 100 segments of 64 bytes; a comma is also allowed; a comma will turn dial tone back on;’+’ is allowed as a valid digit; extension letter ‘R’ is used as defined above. This parameter enables the phone to automatically initiate calls to numbers that match a digit map pattern.		

Parameter	*Permitted Values*	*Default*
dialplan.digitmap.timeOut[1]	**string of positive integers separated by '|'**	**3 | 3 | 3 | 3 | 3| 3**
Specify a timeout in seconds for each segment of digit map. After you press a key, the phone will wait this many seconds before matching the digits to a dial plan and dialing the call. Note: If there are more digit maps than timeout values, the default value of 3 will be used. If there are more timeout values than digit maps, the extra timeout values are ignored.		
dialplan.filterNonDigitUriUsers[1]	**0 or 1**	**0**
If 0, allow do not filter out (+) in the dial plan. If 1, filter out (+) from the dial plan (this is the previous behavior).		
dialplan.impossibleMatchHandling[1]	**0, 1 or 2**	**0**
This parameter applies to digits you enter in dial mode, the dial mode when you tap the New Call softkey. The phone is not in dial mode when you are hot dialing, contact dialing, or call list dialing. If set to 0, the digits entered up to and including the point an impossible match occurred are sent to the server immediately. If set to 1, give reorder tone. If set to 2, allow user to accumulate digits and dispatch call manually with the **Send** soft key. Note that if a call orbit number begins with '**#**' or '*', you need to set this parameter to 2 to retrieve the call using off-hook dialing.		
dialplan.removeEndOfDial[1]	**0 or 1**	**1**
If set to 1, strip trailing # digit from digits sent out.		
dialplan.routing.emergency.outboundIdentity	**10-25 digits, or a SIP, or TEL URI**	**Null**
The identity used to identify your phone when you place an emergency call from your phone. Format should be a 10-25 digit number or a valid SIP, or TEL URI. If using a URI, the full uri will be included verbatim in the P-A-I header.		
dialplan.routing.emergency.x.description[1] **Emergency contact description** **dialplan.routing.emergency.x.server.y**[1] **Emergency server** **dialplan.routing.emergency.x.value** **Emergency URL values**	**string** **positive integer** **SIP URL (single entry)**	**x=1:Emergency, Others: Null** **x=1: 1, others: Null** **x=1: 911, others: Null**
x is the index of the emergency entry description and y is the index of the server associated with emergency entry x. For each emergency entry (index x), one or more server entries (indexes (x,y)) can be configured. x and y must both use sequential numbering starting at 1. `description`: The label or description for the emergency address `server.y`: The index representing the server to use for emergency routing (`dialplan.routing.server.x.address` where x is the index). `value`: The URLs that should be watched for. When the user dials one of the URLs, the call will be directed to the emergency server defined by `address`. **Note:** Blind transfer for 911 (or other emergency calls) may not work if registration and emergency servers are different entities.		
dialplan.routing.server.x.address[1]	**dotted-decimal IP address or hostname**	**Null**
The IP address or hostname of a SIP server that will be used for routing calls. Multiple servers can be listed starting with x=1 to 3 for fault tolerance. **Note:** Blind transfer for 911 (or other emergency calls) may not work if registration and emergency servers are different entities.		

Parameter	*Permitted Values*	*Default*
dialplan.routing.server.x.port[1]	**1 to 65535**	**5060**
The port of a SIP server that will be used for routing calls		
dialplan.routing.server.x.transport[1]	**DNSnaptr, TCPpreferred, UDPOnly, TLS, TCPOnly**	**DNSnaptr**
The dns lookup of the first server to be dialed will be used, if there is a conflict with the others. For example, if `dialplan.routing.server.1.transport="UDPOnly"` and `dialplan.routing.server.2.transport = "TLS"`, then `UDPOnly` is used.		
dialplan.userDial.timeOut	**0 – 99 seconds**	**Generic Profile=0**
This parameter specifies the time in seconds that the phone waits before dialing a number you enter while the phone is on hook. You can apply `dialplan.userDial.timeOut` only when its value is lower than up.IdleTimeOut. Note that you need to restart or reboot to apply changes to this parameter.		
dialplan.x.conflictMatchHandling	**0 or 1**	**Generic Profile=0**
This is the per-registration parameter of `dialplan.conflictMatchHandling`. This parameter takes priority over the general parameter, `dialplan.conflictMatchHandling`.		

[1] Change causes phone to restart or reboot.

Per-registration dial plan configuration is also supported and paramters are listed in the table Per Registration Dial Plan (Digit Map) Parameters. The descriptions for these per-registration parameters are provided in the table Dial Plan (Digit Map) Parameters. Note that the per-registration parameters override the general parameters where x is the registration number (for example, `dialplan.x.applyToTelUriDial` overrides `dialplan.applyToTelUriDial` for registration x).

For the CX500 system, x=1-16.

Per-Registration Dial Plan (Digit Map) Parameters

Parameter	*Permitted Values*	*Default*
dialplan.conflictMatchHandling	**0 or 1**	**Generic Profile=0** **Lync Profile=1**
dialplan.x.applyToCallListDial[1]	**0 or 1**	**1**
dialplan.x.applyToDirectoryDial[1]	**0 or 1**	**0**
dialplan.x.applyToForward	**0 or 1**	**0**
dialplan.x.applyToTelUriDial[1]	**0 or 1**	**1**
dialplan.x.applyToUserDial[1]	**0 or 1**	**1**
dialplan.x.applyToUserSend[1]	**0 or 1**	**1**
dialplan.x.digitmap[1]	**string - max number of characters 2560**	**Null**

Parameter	*Permitted Values*	*Default*
dialplan.x.digitmap.timeOut[1]	**string - max number of characters 100**	**Null**
dialplan.x.e911dialmask	**string - max number of characters 256**	**Null**
dialplan.x.e911dialstring	**string - max number of characters 256**	**Null**
dialplan.x.applyToForward	**0 or 1**	**0**
dialplan.x.impossibleMatchHandling[1]	**0 to 2**	**0**
dialpan.x.lyncDigitmap.timeOut	**0-99 seconds**	**3 seconds**
dialplan.x.originaldigitmap	**string - max number of characters 2560**	**Null**
dialplan.x.removeEndOfDial[1]	**0 or 1**	**1**
dialplan.x.routing.emergency.y.value[1]	**string - max number of characters 64**	**Null**
dialplan.x.routing.emergency.y.server.z[1]	**0 to 3**	**0 For all x, y, and z = 1 to 3**
dialplan.x.routing.server.y.address[1]	**string - max number of characters 256**	**Null**
dialplan.x.routing.server.y.port[1]	**1 to 65535**	**5060**
dialplan.x.routing.server.y.transport[1]	**DNSnaptr, TCPpreferred, UDPOnly, TLS, TCPOnly**	**DNSnaptr**
dialplan.userDial.timeOut	**0 – 99 seconds**	**Generic Profile=0** **Lync Profile=3**

[1] Change causes phone to restart or reboot.

<dir>

This parameter definition includes:

- <broadsoft/> Polycom BroadSoft UC-One directory definitions
- <local/> The local directory definition
- <corp/> The corporate directory definition

<broadsoft/>

Use the parameters listed in the table Polycom BroadSoft UC-One Feature Parameters with the Polycom BroadSoft UC-One directory.

Polycom BroadSoft UC-One Feature Parameters

Parameter	*Permitted Values*	*Default*
dir.broadsoft.xsp.address	**dotted-decimal IP address, hostname or FQDN**	**Null**
Set the IP address or hostname of the Broadsoft directory XSP home address. For example, `host.domain.com or http://xxx.xxx.xxx.xxx`.		
dir.broadsoft.xsp.username	**UTF-8 encoding string**	**Null**
Set the username used to authenticate to the BroadSoft Directory XSP server.		
dir.broadsoft.xsp.password	**UTF-8 encoding string**	**Null**
Set the password used to authenticate to the BroadSoft Directory XSP server.		

<local/>

The table Local Contact Directory Parameters lists parameters you can configure for your local contact directory. The local directory is stored in either device settings or RAM on the phone. The local directory size is limited based on the amount of flash memory in the phone. (Different phone models have variable flash memory.)

When the volatile storage option is enabled, ensure that a properly configured provisioning server that allows uploads is available to store a back-up copy of the directory or its contents will be lost when the phone reboots or loses power.

Local Contact Directory Parameters

Parameter	*Permitted Values*	*Default*
dir.local.contacts.maxNum[1]	**1 to 9999**	**99** **9999**
Maximum number of contacts allowed in the local contact directory.		
dir.local.readonly[1]	**0 or 1**	**0**
If 0, the local contact directory can be edited. If 1, the local contact directory is read-only.		
dir.search.field	**0 or 1**	**0**
If 0, contact directory searches are sorted by contact's last name. If 1, contact directory searches are sorted by first name.		

[1] Change causes phone to restart or reboot.

<corp/>

Use the paramters in the table Corporate Directory Parameters to configure a corporate directory. A portion of the corporate directory is stored in flash memory on the phone. The size is based on the amount of flash memory in the phone. Different phone models have variable flash memory.

Corporate Directory Parameters

Parameter	*Permitted Values*	*Default*
dir.corp.address[1]	**dotted-decimal IP address or hostname or FQDN**	**Null**
The IP address or hostname of the LDAP server interface to the corporate directory. For example, host.domain.com.		
dir.corp.attribute.x.filter[1]	**UTF-8 encoded string**	**Null**
The filter string for this parameter, which is edited when searching.		
dir.corp.attribute.x.label[1]	**UTF-8 encoded string**	**Null**
The label when data is displayed.		
dir.corp.attribute.x.name[1]	**UTF-8 encoded string**	**Null**
The name of the parameter to match on the server. Each name must be unique; however, an LDAP entry can have multiple parameters with the same name. Up to eight parameters can be configured (x = 1 to 8).		
dir.corp.attribute.x.searchable[1]	**0 or 1**	**0**
If 0, quick search on parameter x (if x is 2 or more) is disabled. If 1, quick search on x (if x is 2 or more) is enabled.		
dir.corp.attribute.x.sticky[1]	**0 or 1**	**0**
If 0, the filter criteria for attribute x is reset after a reboot. If 1, the filter criteria are retained through a reboot. If you set an attribute to be sticky (set this parameter to 1), a '*' will display before the label of the attribute on the phone.		
dir.corp.attribute.x.type[1]	**first_name, last_name, phone_number SIP_address, H323_address URL, other**	**last_name**
Defines how parameter x is interpreted by the phone. Entries can have multiple parameters of the same type. The value other is used for display purposes only. If the user saves the entry to the local contact directory on the phone, `first_name`, `last_name`, and `phone_number` are copied. The user can place a call to the `phone_number` and `SIP_address` from the corporate directory.		
dir.corp.autoQuerySubmitTimeout[1]	**0 to 60 seconds**	**0**
The timeout (in seconds) between when the user stops entering characters in the quick search and when the search query is automatically submitted. If 0, there is no timeout (automatic submit is disabled).		
dir.corp.backGroundSync[1]	**0 or 1**	**0**
If 0, background downloading from the LDAP server is disabled. If 1, background downloading is enabled.		

Parameter	*Permitted Values*	*Default*
dir.corp.backGroundSync.period[1]	**3600 to 604800**	**86400**
The corporate directory cache is refreshed after the corporate directory feature has not been used for this period of time seconds. The default period is 24 hours (86400 seconds). The minimum is 1 hour and the maximum is 7 days.		
dir.corp.baseDN[1]	**UTF-8 encoded string**	**Null**
The base domain name. This is the starting point for making queries on the LDAP server.		
dir.corp.bindOnInit[1]	**0 or 1**	**1**
If 0, do not use bind authentication on initialization. If 1, use bind authentication on initialization.		
dir.corp.cacheSize[1]	**8 to 256**	**128**
The maximum number of entries that can be cached locally on the phone.		
dir.corp.filterPrefix[1]	**UTF-8 encoded string**	**(objectclass=person)**
Predefined filter string for search queries.		
dir.corp.pageSize[1]	**8 to 64**	**32**
The maximum number of entries requested from the corporate directory server with each query.		
dir.corp.password[1]	**UTF-8 encoded string**	**Null**
The password used to authenticate to the LDAP server.		
dir.corp.port[1]	**0, Null, 1 to 65535**	**389 (TCP) 636 (TLS)**
The port that connects to the server if a full URL is not provided.		
dir.corp.scope[1]	**one, sub, base**	**sub**
The type of search that is performed. If **one**, a search of one level below the base domain name (DN). If **sub**, a recursive search of all levels below the base DN. If **base**, a search at the base DN level.		
dir.corp.sortControl[1]	**0 or 1**	**0**
Control how a client can make queries and sorts entries locally. If 0, leave sorting as negotiated between the client and server. If 1, force sorting of queries (this causes excessive LDAP queries and should only be used to diagnose LDAP servers with sorting problems).		
dir.corp.transport[1]	**TCP, TLS, Null**	**TCP**
Specify whether a TCP or TLS connection is made with the server, if a full URL is not provided.		
dir.corp.user[1]	**UTF-8 encoded string**	**Null**
The user name used to authenticate to the LDAP server.		
dir.corp.viewPersistence[1]	**0 or 1**	**0**
If 0, the corporate directory search filters and browsing position are reset each time the user accesses the corporate directory. If 1, the search filters and browsing position from the previous session are displayed each time the user accesses the corporate directory.		

Parameter	*Permitted Values*	*Default*
dir.corp.vlv.allow[1]	**0 or 1**	**0**
If 0, virtual view list (VLV) queries are disabled. If 1, VLV queries are enabled and can be made if the LDAP server supports VLV.		
dir.corp.vlv.sortOrder[1]	**list of parameters**	**Null**
The list of parameters —in exact order—for the LDAP server to use when indexing. For example: `sn`, `givenName`, `telephoneNumber`.		

[1] Change causes phone to restart or reboot.

<divert/>

The phone has a flexible call forward/diversion feature for each registration. In all cases, a call will only be diverted if a non-Null contact has been configured.

In the table Call Diversion (Call Forwarding) Parameters, x is the registration number. For CX5500, x=1-16.

Call Diversion (Call Forwarding) Parameters

Parameter	*Permitted Values*	*Default*
divert.x.contact[1]	**contact address: ASCII encoded string containing digits (the user part of a SIP URL) or a string that constitutes a valid SIP URL (6416 or 6416@polycom.com)**	**Null**
The forward-to contact used for all automatic call diversion features. All automatically forwarded calls will be directed to this contact. The contact can be overridden by a busy contact, DND contact, or no-answer contact as specified by the `busy`, `dnd`, and `noAnswer` parameters that follow.		
divert.x.sharedDisabled[1]	**0 or 1**	**1**
If 0, call diversion features can be used on shared lines. If 1, call diversion features are disabled on shared lines.		
divert.x.autoOnSpecificCaller[2]	**0 or 1**	**1**
If 0, the Auto Divert feature of the contact directory is disabled for registration x. If 1, calls on registration x may be diverted using Auto Divert, you may specify to divert individual calls or divert all calls.		
divert.busy.x.enabled[2] **divert.busy.x.contact**[1]	**0 or 1** **contact address**	**1** **Null**
Divert incoming calls that reach a busy signal. If `enabled` is set to 1, calls will be diverted when registration x is busy. Calls will be sent to the busy contact's address if it is specified; otherwise calls will be sent to the default contact specified by `divert.x.contact`. If `enabled` is set to 0, calls will not be diverted if the line is busy.		

Parameter	*Permitted Values*	*Default*
divert.dnd.x.enabled[2]	**0 or 1**	**0**
divert.dnd.x.contact[1]	**contact address**	**Null**
Divert calls when Do Not Disturb is enabled. If `enabled` is set to 1, calls will be diverted when DND is enabled on registration x. Calls will be sent to the DND contact's address if it is specified; otherwise calls will be sent to the default contact specified by `divert.x.contact`.		
divert.fwd.x.enabled[2]	**0 or 1**	**1**
If 0, the user cannot enable universal call forwarding (automatic forwarding for all calls on registration x). If 1, a Forward soft key displays on the phone's Home screen that you can use to enable universal call forwarding.		
divert.noanswer.x.enabled[2]	**0 or 1**	**1**
divert.noanswer.x.contact[1]	**contact address**	**Null**
divert.noanswer.x.timeout[1]	**positive integer**	**55**
If no-answer call diversion is `enabled`, calls that are not answered after the number of seconds specified by timeout will be sent to the no-answer `contact`. If the no-answer `contact` is set to Null, the call will be sent to the default contact specified by `divert.x.contact`. If `enabled` is set to 0, calls will not be diverted if they are not answered.		

[1] Change causes phone to restart or reboot.

[2] Change causes phone to restart or reboot. If server-based call forwarding is enabled, this parameter is disabled.

<dns/>

The <dns/> parameters include:

- DNS-A
- DNS-NAPTR
- DNS-SRV

You can enter a maximum of 12 record entries for DNS-A, DNS-NAPTR, and DNS-SRV records.

DNS-A

Add up to 12 DNS-A record entries using the parameters in the table DNA-A Parameters. Specify the address, name, and cache time interval for DNS-A record *x*, where x is from 1 to 12.

DNA-A Parameters

Parameter	*Permitted values*	*Default*
dns.cache.A.x.address	**dotted-decimal IP version 4 address**	**Null**
IP address.		
dns.cache.A.x.name	**valid hostname**	**Null**
Hostname		

Parameter	*Permitted values*	*Default*
dns.cache.A.x.ttl	**300 to 536870912 (2^29), seconds**	**300**
The TTL describes the time period the phone will use the configured static cache record. If a dynamic network request receives no response, this timer begins on first access of the static record and once the timer expires, the next lookup for that record will retry a dynamic network request before falling back on the static entry and its reset TTL timer again.		

DNS-NAPTR

Add up to 12 DNS-NAPTR record entries using parameters in the table DNS-NAPTR Parameters. Specify each parameter for DNS-NAPTR record *x*, where x is from 1 to 12.

DNS-NAPTR Parameters

Parameter	*Permitted values*	*Default*
dns.cache.NAPTR.x.flags	**A single character from [A-Z, 0-9]**	**Null**
The flags to control aspects of the rewriting and interpretation of the fields in the record. Characters are case-sensitive. At this time, only 'S', 'A', 'U', and 'P' are defined as flags. See RFC 2915 for details of the permitted flags.		
dns.cache.NAPTR.x.name	**domain name string**	**Null**
The domain name to which this resource record refers.		
dns.cache.NAPTR.x.order	**0 to 65535**	**0**
An integer specifying the order in which the NAPTR records must be processed to ensure the correct ordering of rules.		
dns.cache.NAPTR.x.preference	**0 to 65535**	**0**
A 16-bit unsigned integer that specifies the order in which NAPTR records with equal "order" values should be processed. Low numbers are processed before high numbers.		
dns.cache.NAPTR.x.regexp	**string containing a substitution expression**	**Null**
This parameter is currently unused. Applied to the original string held by the client. The substitution expression is applied in order to construct the next domain name that will be looked up. The grammar of the substitution expression is given in RFC 2915.		
dns.cache.NAPTR.x.replacement	**domain name string with SRV prefix**	**Null**
The next name to query for NAPTR records depending on the value of the flags field. It must be a fully qualified domain-name.		
dns.cache.NAPTR.x.service	**string**	**Null**
Specifies the service(s) available down this rewrite path. For more information, see RFC 2915.		

Parameter	*Permitted values*	*Default*
dns.cache.NAPTR.x.ttl	**300 to 536870912 (2^29), seconds**	**300**
The TTL describes the time period the phone will use the configured static cache record. If a dynamic network request receives no response, this timer begins on first access of the static record and once the timer expires, the next lookup for that record will retry a dynamic network request before falling back on the static entry and its reset TTL timer again.		

DNS-SRV

Add up to 12 DNS-SRV record entries using parameters in the table DNS-SRV Parameters. Specify each parameter for DNS-SRV record *x*, where x is from 1 to 12.

DNS-SRV Parameters

Parameter	*Permitted values*	*Default*
dns.cache.SRV.x.name	**domain name string with SRV prefix**	**Null**
The domain name string with SRV prefix.		
dns.cache.SRV.x.port	**0 to 65535**	**0**
The port on this target host of this service. For more information, see RFC 2782.		
dns.cache.SRV.x.priority	**0 to 65535**	**0**
The priority of this target host. For more information, see RFC 2782.		
dns.cache.SRV.x.target	**domain name string**	**Null**
The domain name of the target host. For more information, see RFC 2782.		
dns.cache.SRV.x.ttl	**300 to 536870912 (2^29), seconds**	**300**
The TTL describes the time period the phone will use the configured static cache record. If a dynamic network request receives no response, this timer begins on first access of the static record and once the timer expires, the next lookup for that record will retry a dynamic network request before falling back on the static entry and its reset TTL timer again.		
dns.cache.SRV.x.weight	**0 to 65535**	**0**
A server selection mechanism. For more information, see RFC 2782.		

<efk/>

Use the three tables to configure the Enhanced Feature Key (EFK) feature on your phone:

- Enhanced Feature Key Version Parameters
- Enhanced Feature Key List Parameters

- Enhanced Feature Key Prompt Parameters

Enhanced Feature Key (EFK) Version Parameters

Parameter Name	*Permitted Values*	*Default*
efk.version	**2 (1 for SIP 3.0 and earlier)**	**2**
The version of the EFK elements. For SIP 3.0.x or earlier, **1** is the only supported version. For SIP 3.1 and later, **2** is the only supported version. If this parameter is Null, the EFK feature s disabled. This parameter is not required if there are no `efk.efklist` entries.		

Enhanced Feature Key (EFK) List Parameters

Parameter Name	*Permitted Values*	*Default*
efk.efklist.x.action.string		
The action string contains a macro definition of the action that the feature key will perform. If EFK is enabled, this parameter must have a value (it cannot be Null). For a list of macro definitions and example macro strings, see Understanding Macro Definitions.		
efk.efklist.x.label	**string**	**Null**
The text string that will be used as a label on any user text entry screens during EFK operation. If Null, the Null string is used. Note: If the label does not fit on the screen, the text will be shortened and '…' will be appended.		
efk.efklist.x.mname		**expanded_macro**
The unique identifier used by the speed dial configuration to reference the enhanced feature key entry. Cannot start with a digit. Note that this parameter must have a value, it cannot be Null.		
efk.efklist.x.status	**0 or 1**	**0**
If 0 or Null, key x is disabled. If 1, the key is enabled.		
efk.efklist.x.type		**invite**
The SIP method to be performed. If set to `invite`, the action required is performed using the SIP INVITE method. Note: This parameter is included for backwards compatibility. Do not use if possible. If `efk.x.action.string` contains types, this parameter is ignored. If Null, the default of INVITE is used.		

Enhanced Feature Key (EFK) Prompt Parameters

Parameter Name	*Permitted Values*	*Default*
efk.efkprompt.x.label[1]	**string**	**Null**
The prompt text that is presented to the user on the user prompt screen. If Null, no prompt displays. Note: If the label does not fit on the screen, the label will be shortened and '…' will be appended.		

Parameter Name	*Permitted Values*	*Default*
efk.efkprompt.x.status[1]	**0 or 1**	**0**
If 0, key x is disabled. If 1, the key is enabled. This parameter must have a value, it cannot be Null. Note: If a macro attempts to use a prompt that is disabled or invalid, the macro execution will fail.		
efk.efkprompt.x.type[1]	**numeric or text**	**text**
The type of characters entered by the user. If set to `numeric`, the characters are interpreted as numbers. If set to `text`, the characters are interpreted as letters. If Null, `numeric` is used. If this parameter has an invalid value, this prompt, and all parameters depending on this prompt, are invalid. Note: A mix of `numeric` and `text` is not supported.		
efk.efkprompt.x.userfeedback[1]	**visible or masked**	**visible**
The user input feedback method. If set to `visible`, the text is visible. If set to `masked`, the text displays as asterisk characters (*), this can be used to mask password fields. If Null, visible is used. If this parameter has an invalid value, this prompt, and all parameters depending on this prompt, are invalid.		

[1] Change causes phone to restart or reboot.

<exchange/>

Set the connection parameters for the Microsoft Exchange application to configure the Calendaring feature. Use the table Microsoft Exchange Parameters which lists available parameters.

Microsoft Exchange Parameters

Parameter	*Permitted Values*	*Default*
exchange.meeting.phonePattern	**String**	**Null**
The pattern used to identify phone numbers in meeting descriptions, where "x" denotes any digit and "|" separates alternative patterns (for example, xxx-xxx-xxxx|604.xxx.xxxx).		
exchange.meeting.reminderEnabled	**0 or 1**	**1**
If 0, meeting reminders are disabled. If 1, they are enabled.		
exchange.server.url[1]	**String**	**Null**
The Microsoft Exchange server address.		

[1] Change causes phone to restart or reboot.

<feature/>

The feature parameters listed in the table Feature Activation/Deactivation Parameters control the activation or deactivation of a feature at run time.

Feature Activation/Deactivation Parameters

Parameter	*Permitted Values*	*Default*
feature.acdAgentAvailable.enabled[1]	**0 or 1**	**0**
If 0, the ACD agent available/unavailable feature is disabled. If 1, the feature is enabled.		
feature.acdLoginLogout.enabled[1]	**0 or 1**	**0**
If 0, the ACD login/logout feature is disabled. If 1, the feature is enabled.		
feature.acdPremiumUnavailability.enabled[1]	**0 or 1**	**0**
If 0, the premium ACD unavailability feature is disabled. If 1, premium ACD unavailability feature is enabled, and unavailability reason codes can be used (if the other ACD feature parameters are also be enabled).		
feature.acdServiceControlUri.enabled[1]	**0 or 1**	**0**
If 0, the ACD service control URI feature is disabled. If 1, the feature is enabled.		
feature.broadsoftdir.enabled	**0 or 1**	**0**
If 1, the BroadSoft Enterprise directory is enabled. If 0, the directory is disabled		
feature.broadsoftUcOne.enabled	**0 or 1**	**0**
If 1, the BroadSoft UC-One feature is enabled. If 0, the feature is disabled.		
Feature.btoe.enabled		
If 0, the Better Togeteher over Ethernet feature is disabled. If 1, the feature is enabled.		
feature.callCenterStatus.enabled	**0 or 1**	**0**
If 0, the Status Event Threshold capability is disabled. If 1, the Status Event Threshold capability is enabled.		
feature.callList.enabled[1] **All locally controlled call lists.**	**0 or 1**	**1**
feature.callListMissed.enabled[1] **The missed calls list.**	**0 or 1**	**1**
feature.callListPlaced.enabled[1] **The placed calls list.**	**0 or 1**	**1**
feature.callListReceived.enabled[1] **The received calls list.**	**0 or 1**	**1**
If 0, the call list is disabled. If 1, the call list is enabled. To enable the Missed, Placed, or Received call lists, `feature.callList.enabled` must be enabled.		
feature.callPark.enabled[1]	**0 or 1**	**0**
If 0, the call park and call retrieve features are disabled. If 1, the features are enabled.		

Parameter	*Permitted Values*	*Default*
feature.callRecording.enabled[1]	**0 or 1**	**0**
If 0, the call recording and playback feature is disabled. If 1, the feature is enabled.		
feature.corporateDirectory.enabled	**0 or 1**	**0**
If 0, the corporate directory feature is disabled. If 1, the feature is enabled.		
feature.directedCallPickup.enabled[1]	**0 or 1**	**0**
If 0, the directed call pickup feature is disabled. If 1, the feature is enabled.		
feature.directory.enabled	**0 or 1**	**1**
If 0, the local contact directory is disabled. If 1, the directory is enabled.		
feature.enhancedCallDisplay.enabled	**0 or 1**	**0**
If 0, the phone may display the protocol at the end of the called party identification (for example, 1234567 [SIP]). If 1, the phone will display the number only (for example, 1234567).		
feature.enhancedFeatureKeys.enabled	**0 or 1**	**0**
If 0, the enhanced feature keys feature is disabled. If 1, the feature is enabled.		
feature.exchangeCalendar.enabled[1]	**0 or 1**	**0**
If 0, the calendaring feature is disabled. If 1, the feature is enabled.		
feature.groupCallPickup.enabled[1]	**0 or 1**	**0**
If 0, the group call pickup feature is disabled.		
feature.lastCallReturn.enabled[1]	**0 or 1**	**0**
If 0, the last call return feature is disabled. If 1, the feature is enabled.		
feature.lync.abs.enabled	**0 or 1**	**1**
Set to 1 to enable comprehensive contact search in the Lync Server address book service. Set to 0 to disable comprehensive contact search in the Lync Server address book service.		
feature.lync.abs.maxResult	**5 to 50**	**20**
The value for this parameter defines the maximum number of contacts to display in a Lync Server address book service contact search.		
Feature.lyncbtoe.auto.signin.signoff.enabled		
Enables or disables the phone to signout of Lync automatically when BToE enabled. If 1, the phone signs out of Lync automatically when BTOE is disabled or the phone is unpaired with the computer. If 0, the phone does not signout of Lync when BToE is disabled or the phone is unpaired with the computer.		
feature.messaging.enabled[1]	**0 or 1**	**0**
If 0, the instant messaging feature is disabled. If 1, the feature is enabled.		
feature.nonVolatileRingerVolume.enabled	**0 or 1**	**1**
If 0, user changes to the ringer volume are reset to default when the phone reboots. If 1, user changes to the ringer volume are saved and maintained when the phone reboots.		

Parameter	*Permitted Values*	*Default*
feature.nWayConference.enabled	**0 or 1**	**0**
If 0, the n-way conferencing managing feature is disabled and while three-way conferencing can exist, there is no manage conference page. If 1, n-way conferencing is enabled, conferences with the maximum number of parties are allowed, and the manage conference page is shown.		
feature.presence.enabled[1]	**0 or 1**	**0**
If 0, the presence feature—including buddy managements and user status—is disabled. If 1, the presence feature is enabled with the buddy and status options.		
feature.qml.enabled[1]	**0 or 1**	**0**
If 1, the QML viewer is enabled on phone. If 0, the viewer is disabled. The viewer is used to load the QML applications.		
feature.ringDownload.enabled[1]	**0 or 1**	**1**
If 0, the phone will not download ringtones when it starts up. If 1, the phone will download ringtones when it starts up.		
feature.urlDialing.enabled	**0 or 1**	**1**
If 0, URL/name dialing is not available. If 1, URL/name dialing is available from private lines. Note: If enabled, unknown callers will be identified on the display by their phone’s IP address.		

[1] Change causes phone to restart or reboot.

<httpd/>

The phone contains a local Web Configuration Utility server for user and administrator features. You can disable it for applications when it is not needed or where it poses a security threat. The Web server supports both basic and digest authentication. The authentication user name and password are not configurable for this release. You can configure the parameters listed in the table HTTPD (Web Server) Parameters.

HTTPD (Web Server) Parameters

Parameter	*Permitted Values*	*Default*
httpd.enabled[1]	**0 or 1**	**1**
If 0, the HTTP server is disabled (the Web Configuration Utility will also be disabled). If 1, the server will be enabled.		
httpd.cfg.enabled[1]	**0 or 1**	**1**
If 0, the Web Configuration Utility is disabled. If 1, the Web Configuration Utility is enabled.		
httpd.cfg.port[1]	**1 to 65535**	**80**
Port is 80 for HTTP servers. Care should be taken when choosing an alternate port.		

Parameter	*Permitted Values*	*Default*
httpd.cfg.secureTunnelEnabled[1]	**0 or 1**	**1**
If 0, the Web does not use a secure tunnel. If 1, the server connects through a secure tunnel.		
httpd.cfg.secureTunnelPort[1]	**1 to 65535**	**443**
The port to use for communications when the secure tunnel is used.		
httpd.cfg.secureTunnelRequired[1]	**0 or 1**	**0**
If 0, communications to the Web server do not require a secure tunnel. If 1, communications do require a secure tunnel.		

[1] Change causes phone to restart or reboot.

<keyboard/>

The parameters listed in Keyboard for Lync Server are for use with Lync Server. Use these parameters to set options for the phone screen virtual keyboard layout and encoding options.

Keyboard for Lync Server

Parameter	*Permitted Values*	*Default*
keyboard.layout.type	**0 or 1**	**0**
Use this parameter to change the phone's virtual keyboard for character and numeric input from the English language QWERTY layout to French language AZERTY layout. The default value 0 sets the virtual keyboard to QWERTY. Set to 1 to use the French-language AZERTY layout.		
keyboard.encoding.all	**0 or 1**	**1**
When set to 1, the default, the phone display default character encoding options for the phone menus. Set to 0 to display only ASCII and Latin encoding options for the phone menus.		

<lcl/>

You can configure the language you want the Polycom phone user interface to operate and display in. The phones support both North American and international time and date formats.

Caution: Use a Multilingual XML Editor

Edit the language parameters using a multilingual XML editor. If you do not use an XML editor, some of the language labels in the configuration file, and in the language menu on the phone, will display incorrectly. To confirm whether your editor properly supports these characters, view the language parameter for languages such as Chinese, Japanese, Korean, Russian— for example `lcl.ml.lang.menu.1.label`.

This parameter definition includes:

-
- <ml/> The multilingual definitions
- <datetime/> The date and time definitions

<ml/>

The multilingual parameters listed in the table Multilingual Parameters is based on string dictionary files downloaded from the provisioning server. These files are encoded in standalone XML format and include several eastern European and Asian languages. The files include space for user-defined languages.

Multilingual Parameters

Parameter	*Permitted Values*
lcl.ml.lang	**Null or an exact match for one of the label names stored in lcl.ml.lang.menu.x.label**
If Null, the default internal language (US English) will be used, otherwise, the language to be used may be specified in the format of `lcl.ml.lang.menu.x.label`. For example, to get the phone to boot up in German, set this parameter to `Deutsch (de-de)`.	
lcl.ml.lang.charset[1]	**string**
The language character set.	
lcl.ml.lang.clock.x.24HourClock	**0 or 1**
If parameter present, overrides `lcl.datetime.time.24HourClock` If 1, display time in 24-hour clock mode rather than am/pm.	
lcl.ml.lang.clock.x.dateTop	**0 or 1**
If parameter present, overrides `lcl.datetime.date.dateTop`. If 1, display date above time, otherwise display time above date.	
lcl.ml.lang.clock.x.format	**string which includes ‘D’, ‘d’ and ‘M’ and two optional commas**
If parameter present, overrides `lcl.datetime.date.format`; D = day of week d = day M = month. Up to two commas may be included. For example: D,dM = Thursday, 3 July or Md,D = July 3, Thursday The field may contain 0, 1 or 2 commas which can occur only between characters and only one at a time. For example: “D,,dM” is illegal.	
lcl.ml.lang.clock.x.longFormat	**0 or 1**
If parameter present, overrides `lcl.datetime.date.longFormat`. If 1, display the day and month in long format (Friday/November), otherwise use abbreviations (Fri/Nov).	
lcl.ml.lang.font.x[1]	**string**
The language font.	

Parameter	*Permitted Values*
lcl.ml.lang.list[1]	**a comma-separated list**
A list of the languages supported on the phones.	
lcl.ml.lang.menu.x **Dictionary file** **lcl.ml.lang.menu.x.label[1]** **Phone language menu label**	**String in the format language_region** **String in the format nativeLanguageName (abbreviation)**
The phone supports multiple languages. Dictionary files and labels must be sequential (for example, `lcl.ml.lang.menu.1`, `lcl.ml.lang.menu.2`, `lcl.ml.lang.menu.3`... `lcl.ml.lang.menu.N`) The dictionary file cannot have caps, and the strings must exactly match a folder name of a dictionary file (you can find the names in the **SoundPointIPLocalization** folder of your software distribution). If you edit these parameters, you need to use a multilingual XML editor that supports Unicode, such as XML Notepad 2007. For example, a dictionary file and label for German would be: `lcl.ml.lang.menu.8="German_Germany"` `lcl.ml.lang.menu.8.label="Deutsch (de-de)"`	

[1] Change causes phone to restart or reboot.

To add a new language:

1 Create a new dictionary file based on an existing one.

2 Change the strings making sure to encode the XML file in UTF-8 but also ensuring the UTF-8 characters chosen are within the Unicode character ranges indicated in the tables below.

3 Place the file in an appropriately named folder according to the format `language_region` parallel to the other dictionary files under the SoundPointIPLocalization folder on the provisioning server.

4 Add an `lcl.ml.lang.clock.menu.x` parameter to the configuration file.

5 Add `lcl.ml.lang.clock.x.24HourClock`, `lcl.ml.lang.clock.x.format`, `lcl.ml.lang.clock.x.longFormat`, and `lcl.ml.lang.clock.x.dateTop` parameters and set them according to the regional preferences.

6 (Optional) Set `lcl.ml.lang` to be the new `language_region` string.

The basic character support includes the Unicode character ranges listed in the table Unicode Ranges for Basic Character Support.

Unicode Ranges for Basic Character Support

Name	*Range*
C0 Controls and Basic Latin	U+0000 - U+007F
C1 Controls and Latin-1 Supplement	U+0080 - U+00FF
Cyrillic (partial)	U+0400 - U+045F

<datetime/>

The parameters listed in the table Date and Time Parameters configure the date and time display on the phone.

Date and Time Parameters

Parameter	*Permitted Values*	*Default*
lcl.datetime.date.dateTop	**0 or 1**	
If set to 1, display date above time else display time above date.		
lcl.datetime.date.format	**string which includes ‘D’, ‘d’ and ‘M’ and two optional commas**	
Controls format of date string. D = day of week, d = day, M = month. Up to two commas may be included. For example: `D,dM` = Thursday, 3 July or `Md,D` = July 3, Thursday The field may contain 0, 1 or 2 commas which can occur only between characters and only one at a time. For example: “D,,dM” is illegal.		
lcl.datetime.date.longFormat	**0 or 1**	
If set to 1, display the day and month in long format (Friday/November), otherwise, use abbreviations (Fri/Nov).		
lcl.datetime.time.24HourClock	**0 or 1**	
If set to 1, display time in 24-hour clock mode rather than a.m./p.m.		

<loc/>

The values you enter for the Lync Server-only parameters listed in the table E.911 Services Parameters are used by E.911 services.

E.911 Services Parameters

Parameter	*Permitted Values*	*Default*
locInfo.source	**LLDP=1** **MS_E911_LIS=2** **CONFIG=3**	**Generic Profile=1** **Lync Profile=2**
This parameter specifies the source of location information for the phone and is useful for locating a phone in environments that have multiple sources of location information. When set to LLDP, location information sent from the network switch is used as the current location. When set to MS_E911_LIS, location information sent from Lync Server is used as current location. When set to CONFIG, you can manually configure location information as current location. If location information is not available from a specified default or configured source, the fall back priority is as follows: Generic profile: LLDP > CONFIG > MS_E911_LIS Lync profile : MS_E911_LIS > CONFIG > LLDP		
locInfo.x.label	**String**	**Null**
Enter a label for your location.		
locInfo.x.country	**String**	**Null**
Enter the country the phone is located in.		
locInfo.x.A1	**String**	**Null**
Enter the national subdivision the phone is located in, for example, a state or province.		
locInfo.x.A3	**String**	**Null**
Enter the city the phone is located in.		
locInfo.x.PRD	**String**	**Null**
Enter the leading direction of the street location.		
locInfo.x.RD	**String**	**Null**
The name of the road or street the phone is located on.		
locInfo.x.STS	**String**	**Null**
Enter the suffix of the name used in locInfo.x.RD, for example, Street, Avenue.		
locInfo.x.POD	**String**	**Null**
Enter the trailing street direction, for example SW.		
locInfo.x.HNO	**String**	**Null**
Enter the street address number of the phone's location.		

Parameter	*Permitted Values*	*Default*
locInfo.x.HNS	**String**	**Null**
Enter a suffix for the street address used in locInfo.x.HNS, for example, ^{A} or ½.		
locInfo.x.LOC	**String**	**Null**
Enter any additional information that identifies the location.		
locInfo.x.NAM	**String**	**Null**
Enter a name for the location, for example, a business name, an occupant, a resident.		
locInfo.x.PC	**String**	**Null**
Enter the postal code of the location.		

<log/>

The event logging system supports the classes of events listed in the table Logging Levels. Two types of logging are supported:

- <level/> <change/>and<render/>
- <sched/>

Caution: Changing the Logging Parameters

Logging parameter changes can impair system operation. Do not change any logging parameters without prior consultation with Polycom Technical Support.

Logging Levels

Logging Level	*Interpretation*
0	Debug only
1	High detail class event
2	Moderate detail event class
3	Low detail event class
4	Minor error—graceful recovery
5	Major error—will eventually incapacitate the system
6	Fatal error

Each event in the log contains the following fields separated by the | character:

- time or time/date stamp
- 1-5 character component identifier (such as "so")
- event class
- cumulative log events missed due to excessive CPU load
- free form text - the event description

Three formats available for the event timestamp are listed in the table Event Timestamp Formats.

Event Timestamp Formats

Type	*Example*
0 - seconds.milliseconds	011511.006 -- 1 hour, 15 minutes, 11.006 seconds since booting.
1 - absolute time with minute resolution	0210281716 -- 2002 October 28, 17:16
2 - absolute time with seconds resolution	1028171642 -- October 28, 17:16:42

<level/> <change/>and<render/>

This configuration parameter is defined in the table Logging Level Change and Render Parameters.

Logging Level, Change, and Render Parameters

Parameter	*Permitted Values*	*Default*
log.level.change.xxx	**0 to 6**	**4**
Control the logging detail level for individual components. These are the input filters into the internal memory-based log system. Possible values for xxx are acom, ares, app1, bluet, bdiag, brow, cap, cdp, cert, cfg, cipher, clink, clist, cmp, cmr, copy, curl, daa, dbs, dbuf, dhcpc, dis, dock, dot1x, dns, drvtbt, ec, efk, ethf, h323, hset, httpa, httpd, hw, ht, ib, key, ldap, lic, lldp, loc, log, mb, mobil, net, niche, oaip, ocsp, osd, pcap, pcd, pdc, peer, pgui, pmt, pnetm, poll, pps, pres, pstn, push, pwrsv, rdisk, res, rtos, rtls, sec, sig, sip, slog, so, soem, srtp, sshc, ssps, style, sync, sys, ta, task, tls, trace, ttrs, usb, usbio, util, utilm, wdog, wlan, wmgr, and xmpp.		
log.render.file	**0 or 1**	**1**
Set to 1. Polycom recommends that you do not change this value.		
log.render.file.size	**positive integer, 1 to 180**	**32**
Maximum size of flash memory for logs in Kbytes. When this size is about to be exceeded, the phone will upload all logs that have not yet been uploaded, and erase half of the logs on the phone. The administrator may use Web browser to read all logs on the phone.		
log.render.file.upload.append	**0 or 1**	**1**
If set to 1, use append mode when uploading log files to server. Note: HTTP and TFTP don't support append mode unless the server is set up for this.		

Parameter	*Permitted Values*	*Default*
log.render.file.upload.append.limitMode	**delete, stop**	**delete**
Behavior when server log file has reached its limit. delete=delete file and start over stop=stop appending to file		
log.render.file.upload.append.sizeLimit	**positive integer**	**512**
Maximum log file size that can be stored on provisioning server in Kbytes.		
log.render.file.upload.period	**positive integer**	**172800**
Time in seconds between log file uploads to the provisioning server. Note: The log file will not be uploaded if no new events have been logged since the last upload.		
log.render.level	**0 to 6**	**1**
Specifies the lowest class of event that will be rendered to the log files. This is the output filter from the internal memory-based log system. The log.render.level maps to syslog severity as follows: 0 -> SeverityDebug (7) 1 -> SeverityDebug (7) 2 -> SeverityInformational (6) 3 -> SeverityInformational (6) 4 -> SeverityError (3) 5 -> SeverityCritical (2) 6 -> SeverityEmergency (0) For more information, refer to Syslog Menu.		
log.render.realtime	**0 or 1**	**1**
Set to 1. Polycom recommends that you do not change this value.		
log.render.stdout	**0 or 1**	**1**
Set to 1. Polycom recommends that you do not change this value.		
log.render.type	**0 to 2**	**2**
Refer to Event Timestamp Formats for timestamp type.		

<sched/>

The phone can be configured to schedule certain advanced logging tasks on a periodic basis. Polycom recommends that you set the parameters listed in the table Logging Schedule Parameters in consultation with Polycom Technical Support. Each scheduled log task is controlled by a unique parameter set starting with log.sched.x where *x* identifies the task. A maximum of 10 schedule logs is allowed.

Logging Schedule Parameters

Parameter	*Permitted Values*
log.sched.x.level	**0 to 5, default 3**
Event class to assign to the log events generated by this command. This needs to be the same or higher than log.level.change.slog for these events to display in the log.	
log.sched.x.name	**alphanumeric string**
Name of an internal system command to be periodically executed. To be supplied by Polycom.	
log.sched.x.period	**positive integer, default 15**
Seconds between each command execution. 0=run once	
log.sched.x.startDay	**0 to 7**
When startMode is abs, specifies the day of the week to start command execution. 1=Sun, 2=Mon, ..., 7=Sat	
log.sched.x.startMode	**abs, rel**
Start at an absolute time or relative to boot.	
log.sched.x.startTime	**positive integer OR hh:mm**
Seconds since boot when startMode is rel or the start time in 24-hour clock format when startMode is abs.	

<msg/>

The table Message Waiting Parameters lists parameters you can use to configure message-waiting which is supported on a per-registration basis.

In the following table, x is the registration number. For the CX5500, x=1-16.

Message Waiting Parameters

Parameter	*Permitted Values*	*Default*
msg.bypassInstantMessage[1]	**0 or 1**	**0**
This parameter determines what is shown on the phone menu when you press the **Messages** key. If 0, the phone shows Message Center and Instant Messages. If 1, the phone bypasses these menus and goes directly to voicemail.		
msg.mwi.x.subscribe	**ASCII encoded string containing digits (the user part of a SIP URL) or a string that constitutes a valid SIP URL (6416 or 6416@polycom.com)**	**Null**
If non-Null, the phone will send a SUBSCRIBE request to this contact after boot-up.		

Parameter	*Permitted Values*	*Default*
msg.mwi.x.callBackMode	**contact, registration, disabled**	**registration**
The message retrieval mode and notification for registration x. `contact`: a call is placed to the contact specified by `msg.mwi.x.callback`. `registration`: the registration places a call to itself (the phone calls itself). `disabled`: message retrieval and message notification are disabled.		
msg.mwi.x.callBack	**ASCII encoded string containing digits (the user part of a SIP URL) or a string that constitutes a valid SIP URL (6416 or 6416@polycom.com)**	**Null**
The contact to call when retrieving messages for this registration if `msg.mwi.x.callBackMode` is set to `contact`.		
msg.mwi.x.led	**0, 1**	**1**
Where x is an integer referring to the registration indexed by reg.x. If set to 0, the red MWI LED will **not** flash when there are new unread messages for the selected line. When set to 1, the LED will flash as long as there are new unread voicemail messages *for any line* in which this is parameter is enabled.		

[1] Change causes phone to restart or reboot.

<mwi/>

The parameters listed in the table Message Waiting Indicator Parameters enable and disable a back light on the phone screen to illuminate when you receive a new voicemail message.

Message Waiting Indicator Parameters

Parameter	*Permitted Values*	*Default*
mwi.backLight.disable	**0 or 1**	**0**
A back light on the phone screen illuminates when you receive a new voicemail. Set to 0 to disable the back light message alert. Set to 1 to enable. The default is disabled. If mwi.backLight.disable is set to true then backLight will not be illuminated on new voice message arrival. By default it will be set to false and does not have any impact on existing functionality.		

<nat/>

The parameters listed in the table Network Access Translation Parameters define port and IP address changes used in NAT traversal. The port changes will change the port used by the phone, while the IP entry simply changes the IP advertised in the SIP signaling. This allows the use of simple NAT devices that can redirect traffic, but does not allow for port mapping. For example, port 5432 on the NAT device can be sent to port 5432 on an internal device, but not to port 1234.

Network Access Translation Parameters

Parameter	*Permitted Values*	*Default*
nat.ip[1]	**dotted- decimal IP address**	**Null**
IP address to advertise within SIP signaling - should match the external IP address used by the NAT device.		
nat.keepalive.interval	**0 to 3600**	**0**
The keep-alive interval in seconds. Sets the interval at which phones will send a keep-alive packet to the gateway/NAT device to keep the communication port open so that NAT can continue to function. If Null or 0, the phone will not send out keep-alive messages.		
nat.mediaPortStart[1]	**0 to 65440**	**0**
The initially allocated RTP port. Overrides the value set for `tcIpApp.port.rtp.mediaPortRangeStart`.		
nat.signalPort[1]	**1024 to 65535**	**0**
The port used for SIP signaling. Overrides `voIpProt.local.port`.		

[1] Change causes phone to restart or reboot.

<phoneLock/>

The parameters listed in the table Phone Lock Parameters Enhanced Feature Key feature must be enabled if you want to use the **Lock** soft key.

Phone Lock Parameters

Parameter	*Permitted Values*	*Default*
phoneLock.authorized.x.description	**String**	
The name or description of an authorized number		
phoneLock.authorized.x.value	**string**	
The number or address for an authorized contact		
Up to five (x=1 to 5) authorized contacts that a user can call while their phone is locked. Each contact needs a description to display on the screen, and a phone number or address value for the phone to dial.		
phoneLock.browserEnabled	**0 or 1**	**0**
If 0, the microbrowser or browser is not displayed while the phone is locked. If 1, the microbrowser or browser is displayed while the phone is locked.		
phoneLock.dndWhenLocked	**0 or 1**	**0**
If 0, the phone can receive calls while it is locked. If 1, the phone enters Do-Not-Disturb mode while it is locked. Note: The user can change this setting from the phone user interface.		

Parameter	Permitted Values	Default
phoneLock.enabled[1]	**0 or 1**	**0**
If 0, the phone lock feature is disabled. If 1, the phone lock feature is enabled. Note: To ‘unlock’ the phone remotely (in conjunction with deleting/modifying the overrides files), disable and re-enable this parameter.		
phoneLock.idleTimeout	**0 to 65535**	**0**
The amount of time (in seconds) the phone can be idle before it automatically locks. If 0, automatic locking is disabled.		
phoneLock.lockState	**0 or 1**	**0**
The value for this parameter indicates whether the phone is locked or unlocked and changes each time you lock or unlock the phone. If 0, the phone is unlocked. If 1, the phone is locked. Note that the phone stores and uploads the value each time it changes via the `MAC-phone.cfg`. You can set this parameter remotely using the Web Configuration Utility.		
phoneLock.powerUpUnlocked	**0 or 1**	**0**
Use this parameter to override `phoneLock.lockState`. If 0, the phone retains the value in `phoneLock.lockState`. If 1, you can restart, reboot, or power cycle the phone to override the value for `phoneLock.lockState` in the `MAC-phone.cfg` and start the phone in an unlocked state. You can then lock or unlock the phone locally. Polycom recommends that you do not leave this parameter enabled.		

[1] Change causes phone to restart or reboot.

<powerSaving/>

The power saving feature automatically turns off the phone’s LCD display when not in use.

Power Saving Parameters

Parameter	Permitted Values	Default
powerSaving.enable	**0 or 1**	**1**
If 0, the LCD power saving feature is disabled. If 1, the feature is enabled. The power-saving feature is enabled by default.		
powerSaving.idleTimeout.offHours	**1 to 10**	**1**
The number of minutes to wait while the phone is idle during off hours before activating power saving.		
powerSaving.idleTimeout.officeHours	**1 to 600 minutes**	**480**
The number of minutes to wait while the phone is idle during office hours before activating power saving. Note that the default time is 480 minutes.		
powerSaving.idleTimeout.userInputExtension	**1 to 20**	**10**
The minimum number of minutes to wait while the phone is idle—after using the phone—before activating power saving.		

Parameter	*Permitted Values*	*Default*
powerSaving.officeHours.duration.monday **powerSaving.officeHours.duration.tuesday** **powerSaving.officeHours.duration.wednesday** **powerSaving.officeHours.duration.thursday** **powerSaving.officeHours.duration.friday** **powerSaving.officeHours.duration.saturday** **powerSaving.officeHours.duration.sunday**	**0 to 24** **0 to 24** **0 to 24** **0 to 24** **0 to 24** **0 to 24** **0 to 24**	**12** **12** **12** **12** **12** **0** **0**
The duration of the day's office hours.		
powerSaving.officeHours.startHour.xxx	**0 to 23**	**7**
The starting hour for the day's office hours, where xxx is one of `monday`, `tuesday`", `wednesday`, `thursday`, `friday`, `saturday`, and `sunday` (refer to `powerSaving.officeHours.duration` for an example.		
powerSaving.userDetectionSensitivity.offHours	**0 to 10**	**2**
The sensitivity of the algorithm used to detect the presence of the phone's user during off hours. 10 is the most sensitive. If set to 0, this feature is disabled. The default value was chosen for good performance in a typical office environment and is biased for difficult detection during off hours.		
powerSaving.userDetectionSensitivity.officeHours	**0 to 10**	**7**
The sensitivity of the algorithm used to detect the presence of the phone's user during office hours. 10 is the most sensitive. If set to 0, this feature is disabled. The default value was chosen for good performance in a typical office environment and is biased for easy detection during office hours.		

<pres/>

The table Presence Parameters lists parameters you can configure for the presence feature. Note that the parameter `pres.reg` is the line number used to send SUBSCRIBE. If this parameter is missing, the phone will use the primary line to send SUBSCRIBE.

Presence Parameters

Parameter	*Permitted Values*	*Default*
pres.idleSoftkeys	**0 or 1**	**1**
If 0, the **MyStat** and **Buddies** presence idle soft keys do not display. If 1, the soft keys display.		
pres.idleTimeout.offHours.enabled	**0 or 1**	**1**
If 0, the off hours idle timeout feature is disabled. If 1, the feature is enabled.		
pres.idleTimeout.offHours.period	**1 to 600**	**15**
The number of minutes to wait while the phone is idle during off hours before showing the Away presence status.		
pres.idleTimeout.officeHours.enabled	**0 or 1**	**1**
If 0, the office hours idle timeout feature is disabled. If 1, the feature is enabled.		

Parameter	*Permitted Values*	*Default*
pres.idleTimeout.officeHours.period	**1 to 600**	**15**
The number of minutes to wait while the phone is idle during office hours before showing the Away presence status.		
pres.reg	**1 to 34**	**1**
The valid line/registration number that is used for presence. This registration sends a SUBSCIRBE for presence. If the value is not a valid registration, this parameter is ignored.		

<prov/>

The parameters listed in the table Provisioning Parameters control the provisioning server system for your phones.

Provisioning Parameters

Parameter	*Permitted Values*	*Default*
prov.configUploadPath	**string**	**Null**
The directory - relative to the provisioning server - where the phone uploads the current configuration file when the user selects Upload Configuration. If set to Null, use the provisioning server directory.		
prov.login.automaticLogout	**0 to 46000**	**0**
The time (in minutes) before a non-default user is automatically logged out of the handset. If 0, the user is not automatically logged out.		
prov.login.defaultPassword	**String**	**Null**
The login password for the default user.		
prov.login.defaultOnly	**0 or 1**	**0**
If 1, the default user is the only user who can log in. If 0, other users can log in.		
prov.login.defaultUser	**String**	**Null**
The username for the default user. If present, the user is automatically logged in when the phone boots up and logged in after another user logs out.		
prov.login.enabled	**0 or 1**	**0**
If 0, the user profile feature is disabled. If 1, the user profile feature is enabled.		
prov.login.lcCache.domain	**0 to 64**	**Null**
The user's sign-in domain name.		
prov.login.lcCache.user	**0 to 64**	**Null**
The user's sign-in user name.		

Parameter	*Permitted Values*	*Default*
prov.login.localPassword	**String**	**123**
The password used to validate the user login. It is stored either as plain text or encrypted (an SHA1 hash).		
prov.login.persistent	**0 or 1**	**0**
If 0, users are logged out if the handset reboots. If 1, users remain logged in when the phone reboots.		
prov.login.required	**0 or 1**	**0**
If 1, a user must log in when the login feature is enabled. If 0, the user does not have to log in.		
prov.loginCredPwdFlushed.enabled	**0 or 1**	**1**
If 1, when a user logs in or logs out, the login credential password is reset. If 0, the login credential password is not reset.		
prov.polling.enabled	**0 or 1**	**0**
If 0, the provisioning server is not automatically polled for upgrades. If 1, the provisioning server is polled.		
prov.polling.mode	**abs, rel, random**	**abs**
The polling mode. `abs` The phone polls every day at the time specified by `prov.polling.time`. `rel` The phone polls after the number of seconds specified by `prov.polling.period`. `random` The phone polls at random between a starting time set in `prov.polling.time` and an end time set in `prov.polling.timeRandomEnd`. Note that if you set the polling period in `prov.polling.period` to a time greater than 86400 seconds (one day) polling occurs on a random day within that polling period (meaning values such as 86401 would be over 2 days) and only between the start and end times. The day within the period is decided based upon the phones mac address and will not change with a reboot whereas the time within the start and end is calculated again with every reboot.		
prov.polling.period	**integer > 3600**	**86400**
The polling period in seconds. The polling period is rounded up to the nearest number of days in absolute and random mode. In relative mode, the polling period starts once the phone boots. In random mode, if this is set to a time greater than 86400 (one day) polling occurs on a random day based on the phone's MAC address.		
prov.polling.time	**hh:mm**	**03:00**
The polling start time. Used in absolute and random modes.		
prov.polling.timeRandomEnd	**hh:mm**	**Null**
The polling stop time. Only used in random mode.		
prov.quickSetup.enabled	**0 or 1**	**0**
If 0, the quick setup feature is disabled. If 1, the quick setup feature is enabled.		
prov.startupCheck.enabled	**0 or 1**	**1**
If 0, the phone is not provisioned at startup. If 1, the phone is provisioned at start up. All configuration files, licenses, and overrides are downloaded even if the software changes. (The previous behavior was to reboot as soon as the phone determined that software changed.)		

[1] Change causes phone to restart or reboot.

<ptt/>

The parameters in the table Group Paging Parameters apply to page mode.

Group Paging Parameters

Parameter	*Permitted Values*	*Default*
ptt.address	**multicast IP address**	**224.0.1.116**
The multicast IP address to send page audio to and receive page audio from.		
ptt.pageMode.enable	**0 or 1**	**0**
If 0, group paging is disabled. If 1, group paging is enabled.		
ptt.pageMode.displayName	**up to 64 octet UTF-8 string**	**PTT**
This display name is shown in the caller ID field of outgoing group pages. If Null, the value from `reg.1.displayName` will be used.		
ptt.pageMode.group.x.available **Make the group available to the user**	**0 or 1**	**1**
ptt.pageMode.group.x.allowTransmit **Allow outgoing announcements to the group**	**0 or 1**	**1**
ptt.pageMode.group.x.label **The label to identify the group**	**string**	**ch24: Priority, ch25: Emergency, others: Null**
ptt.pageMode.group.x.subscribed **Subscribe the phone to the group**	**0 or 1**	**ch1, 24, 25: 1, others: 0**
A page mode group x, where x= 1 to 25. The `label` is the name used to identify the group during pages. If `available` is disabled (0), the user cannot access the group or subscribe and the other page mode group parameters will be ignored. If enabled, the user can access the group and choose to subscribe. If `allowTransmit` is disabled (0), the user cannot send outgoing pages to the group. If enabled, the user may send outgoing pages. If `subscribed` is disabled, the phone will not be subscribed to the group. If enabled, the phone will subscribe to the group.		

<qos/>

These parameters listed in the table Quality of Service (Type of Service) Parameters control the following Quality of Service (QoS) options:

- The 802.1p/Q user_priority field RTP, call control, and other packets
- The “type of service” field RTP and call control packets

Quality of Service (Type-of-Service) Parameters

Parameter	*Permitted Values*	*Default*
qos.ethernet.callControl.user_priority[1]	**0 to 7**	**5**
User-priority used for call control packets.		
qos.ethernet.other.user_priority[1]	**0 to 7**	**2**
User-priority used for packets that do not have a per-protocol setting.		
qos.ethernet.rtp.user_priority[1]	**0 to 7**	**5**
Choose the priority of voice Real-Time Protocol (RTP) packets. The default priority level is 5.		
qos.ethernet.rtp.video.user_priority[1]	**0 to 7**	**5**
User-priority used for Video RTP packets.		
qos.ip.callControl.dscp[1]	**0 to 63 or EF or any of AF11,AF12, AF13,AF21, AF22,AF23, AF31,AF32, AF33,AF41, AF42,AF43**	**Null**
Specify the DSCP of packets. If the value is not null, this parameter will override the other `qos.ip.callControl.*` parameters. The default value is Null, so the other `qos.ip.callControl.*` parameters will be used if no value is entered.		
qos.ip.callControl.max_reliability[1] **qos.ip.callControl.max_throughput[1]** **qos.ip.callControl.min_cost[1]** **qos.ip.callControl.min_delay[1]** **qos.ip.callControl.precedence[1]**	**0 or 1** **0 or 1** **0 or 1** **0 or 1** **0 -7**	**0** **0** **0** **1** **5**
Set the bits in the IP ToS field of the IP header used for call control. Specify whether or not to set the max reliability bit, the max throughput bit, the min cost bit, the min delay bit, and the precedence bits. If 0, the bit in the IP ToS field of the IP header is not set. If 1, the bit is set.		
qos.ip.rtp.dscp[1]	**0 to 63 or EF or any of AF11,AF12, AF13,AF21, AF22,AF23, AF31,AF32, AF33,AF41, AF42,AF43**	**Null**
Specify the DSCP of packets. If the value is not null, this parameter will override the other `qos.ip.rtp.*` parameters. The default value is Null, so the other `quality.ip.rtp.*` parameters will be used.		
qos.ip.rtp.max_reliability[1] **qos.ip.rtp.max_throughput[1]** **qos.ip.rtp.min_cost[1]** **qos.ip.rtp.min_delay[1]** **qos.ip.rtp.precedence[1]**	**0 or 1** **0 or 1** **0 or 1** **0 or 1** **0 -7**	**0** **1** **0** **1** **5**
Set the bits in the IP ToS field of the IP header used for RTP. Specify whether or not to set the max reliability bit, the max throughput bit, the min cost bit, the min delay bit, and the precedence bit. If 0, the bit in the IP ToS field of the IP header is not set. If 1, the bit is set.		

Parameter	*Permitted Values*	*Default*
qos.ip.rtp.video.dscp[1]	**0 to 63 or EF or any of AF11,AF12, AF13,AF21, AF22,AF23, AF31,AF32, AF33,AF41, AF42,AF43**	**Null**
Allows the DSCP of packets to be specified. If the value is non-null, this parameter will override the other `qos.ip.rtp.video.*`parameters. The default value is Null, so the other `qos.ip.rtp.video.*` parameters will be used.		
qos.ip.rtp.video.max_reliability[1]	**0 or 1**	**0**
qos.ip.rtp.video.max_throughput[1]	**0 or 1**	**1**
qos.ip.rtp.video.min_cost[1]	**0 or 1**	**0**
qos.ip.rtp.video.min_delay[1]	**0 or 1**	**1**
qos.ip.rtp.video.precedence[1]	**0 -7**	**5**
Set the bits in the IP ToS field of the IP header used for RTP video. Specify whether or not to set the max reliability bit, the max throughput bit, the min cost bit, the min delay bit, and the precedence bit. If 0, the bit in the IP ToS field of the IP header is not set. If 1, the bit is set.		

[1] Change causes phone to restart or reboot.

<reg/>

Each registration can optionally be associated with a private array of servers for completely segregated signaling. The CX5500 system supports a total of 16 registrations.

In the following tables, x is the registration number. For the CX5500, x=1-16.

The tables Registration Parameters and Registration Server Parameters list all line registration and server registration parameters.

Registration Parameters

Parameter	*Permitted Values*	*Default*
reg.x.acd-login-logout	**0 or 1**	**0**
reg.x.acd-agent-available	**0 or 1**	**0**
If both ACD login/logout and agent available are set to 1 for registration x, the ACD feature will be enabled for that registration.		
reg.x.address	**string address**	**Null**
The user part (for example, 1002) or the user and the host part (for example, 1002@polycom.com) of the registration SIP URI extension.		
reg.x.applyServerDigitMapLocally	**0 or 1**	**0**
If 1 and `reg.x.server.y.specialInterop` is set to `lync2010`, the phone uses the dialplan from the Microsoft Lync Server. Any dialed number will apply the dial plan locally. If 0, the dialplan from the Microsoft Lync Server is not used.		

Parameter	*Permitted Values*	*Default*
reg.x.auth.domain	**string**	**Null**
The domain of the authorization server that is used to check the user names and passwords.		
reg.x.auth.optimizedInFailover	**0 or 1**	**0**
The destination of the first new SIP request when failover occurs. If 0, the SIP request is sent to the server with the highest priority in the server list. If 1, the SIP request is sent to the server which sent the proxy authentication request.		
reg.x.auth.password	**string**	**Null**
The password to be used for authentication challenges for this registration. If the password is non-Null, it will override the password entered into the Authentication submenu on the Settings menu of the phone.		
reg.x.auth.userId	**string**	**Null**
User ID to be used for authentication challenges for this registration. If the User ID is non-Null, it will override the user parameter entered into the Authentication submenu on the Settings menu of the phone.		
reg.x.auth.useLoginCredentials	**0 or 1**	**0**
If 0, login credentials are not used for authentication to the server on registration x. If 1, login credentials are used for authentication to the server.		
reg.x.bargeInEnabled	**0 or 1**	**0**
If 0, barge-in is disabled for line x. If 1, barge-in is enabled (remote users of shared call appearances can interrupt or barge in to active calls.		
reg.x.callsPerLineKey[1]	**1-24**	**24**
Set the maximum number of concurrent calls for a single registration x. This parameter applies to all line keys using registration x. If registration x is a shared line, an active call counts as a call appearance on all phones sharing that registration. This parameter overrides `call.callsPerLineKey`.		
reg.x.csta	**0 or 1**	**0**
If 0, the uaCSTA (User Agent Computer Supported Telecommunications Applications) feature is disabled. If 1, uaCSTA is enabled (overrides the global parameter `voIpProt.SIP.csta`.		
reg.x.dialPlanName	**String**	**Null**
If `reg.x.server.y.specialInterop` is set to `lync2010`, the dialplan name from the Microsoft Lync Server is stored here. Each registration has its own name for this dialplan. Note: Do not change this parameter if set by Microsoft Lync.		
reg.x.displayName	**UTF-8 encoded string**	**Null**
The display name used in SIP signaling or as the default caller ID.		
reg.x.filterReflectedBlaDialogs	**0 or 1**	**1**
If 0, bridged line appearance NOTIFY messages (dialog state change) will not be ignored. If 1, the messages will be ignored.		
reg.x.fwd.busy.contact	**string**	**Null**
The forward-to contact for calls forwarded due to busy status. If Null, the contact specified by `divert.x.contact` will be used.		

Parameter	*Permitted Values*	*Default*
reg.x.fwd.busy.status	**0 or 1**	**0**
If 0, incoming calls that receive a busy signal will not be forwarded. If 1, busy calls are forwarded to the contact specified by `reg.x.fwd.busy.contact`.		
reg.x.fwd.noanswer.contact	**string**	**Null**
The forward-to contact used for calls forwarded due to no answer. If Null, the contact specified by `divert.x.contact` will be used.		
reg.x.fwd.noanswer.ringCount	**0 to 65535**	**0**
The number of seconds the phone should ring for before the call is forwarded because of no answer. Note: The maximum value accepted by some call servers is 20.		
reg.x.fwd.noanswer.status	**0 or 1**	**0**
If 0, calls are not forwarded if there is no answer. If 1, calls are forwarded to the contact specified by `reg.x.noanswer.contact` after ringing for the length of time specified by `reg.x.fwd.noanswer.ringCount`.		
reg.x.ice.turn.callAdmissionControl.enabled	**0 or 1**	**0**
If 0, call admission control is disabled. If 1, call admission control is enabled for calls using the Microsoft Lync 2010 Server.		
reg.x.label	**UTF-8 encoded string**	**Null**
The text label that displays next to the line key for registration x. If Null, the user part of `reg.x.address` is used.		
reg.x.lcs	**0 or 1**	**0**
If 0, the Microsoft Live Communications Server (LSC) is not supported for registration x. If 1, LSC is supported.		
reg.x.lineKeys	**1 to max**	**1**
Specify the number of line keys to use for a single registration. The maximum number of line keys you can use per registration depends on your phone model. To find out the maximum number for your phone, see **Error! Reference ource not found.**.		
reg.x.lisdisclaimer	**string, 0 to 256 characters**	**Null**
This parameter sets the value of the location policy disclaimer. For example, the disclaimer may be "Warning: If you do not provide a location, emergency services may be delayed in reaching your location should you need to call for help." This parameter is set by in-band provisioning when the phone is registered to Microsoft Lync Server 2010.		
reg.x.lync.autoProvisionCertLocation	**0 to 6**	**6**
If 0, the certificate download is disabled. If non-0, the certificate corresponding to the index of the appropriate `sec.TLS.customCaCert.X` is downloaded.		
reg.x.musicOnHold.uri	**a SIP URI**	**Null**
A URI that provides the media stream to play for the remote party on hold. If present and not Null, this parameter overrides `voIpProt.SIP.musicOnHold.uri`.		

Parameter	Permitted Values	Default
reg.x.outboundProxy.address	**dotted-decimal IP address or hostname**	**Null**
The IP address or hostname of the SIP server to which the phone sends all requests.		
reg.x.outboundProxy.failOver.failBack.mode	**newRequests** **DNSTTL** **registration** **duration**	**newRequests**
The mode for failover failback (overrides `reg.x.server.y.failOver.failBack.mode`). `newRequests` all new requests are forwarded first to the primary server regardless of the last used server. `DNSTTL` the phone tries the primary server again after a timeout equal to the DNS TTL configured for the server that the phone is registered to. `registration` the phone tries the primary server again when the registration renewal signaling begins. `duration` the phone tries the primary server again after the time specified by `reg.x.outboundProxy.failOver.failBack.timeout` expires.		
reg.x.outboundProxy.failOver.failBack.timeout	**0, 60 to 65535**	**3600**
The time to wait (in seconds) before failback occurs (overrides `reg.x.server.y.failOver.failBack.timeout`).If the fail back mode is set to Duration, the phone waits this long after connecting to the current working server before selecting the primary server again. If 0, the phone will not fail-back until a fail-over event occurs with the current server.		
reg.x.outboundProxy.failOver.failRegistrationOn	**0 or 1**	**0**
When set to 1, and the reRegisterOn parameter is enabled, the phone will silently invalidate an existing registration (if it exists), at the point of failing over. When set to 0, and the reRegisterOn parameter is enabled, existing registrations will remain active. This means that the phone will attempt failback without first attempting to register with the primary server to determine if it has recovered. Note that `reg.x.outboundProxy.failOver.RegisterOn` must be enabled.		
reg.x.outboundProxy.failOver.onlySignalWithRegistered	**0 or 1**	**1**
When set to 1, and the reRegisterOn and failRegistrationOn parameters are enabled, no signaling is accepted from or sent to a server that has failed until failback is attempted or failover occurs. If the phone attempts to send signaling associated with an existing call via an unregistered server (for example, to resume or hold a call), the call will end. No SIP messages will be sent to the unregistered server. When set to 0, and the reRegisterOn and failRegistrationOn parameters are enabled, signaling will be accepted from and sent to a server that has failed (even though failback hasn't been attempted or failover hasn't occurred).		
reg.x.outboundProxy.failOver.reRegisterOn	**0 or 1**	**0**
This parameters overrides `reg.x.server.y.failOver.failBack.RegisterOn`. When set to 1, the phone will attempt to register with (or via, for the outbound proxy scenario), the secondary server. If the registration succeeds (a 200 OK response with valid expires), signaling will proceed with the secondary server. When set to 0, the phone won't attempt to register with the secondary server, since the phone will assume that the primary and secondary servers share registration information.		
reg.x.outboundProxy.port	**1 to 65535**	**0**
The port of the SIP server to which the phone sends all requests.		

Parameter	*Permitted Values*	*Default*
reg.x.outboundProxy.transport	**DNSnaptr, TCPpreferred, UDPOnly, TLS, TCPOnly**	**DNSnaptr**
The transport method the phone uses to communicate with the SIP server. `Null` or `DNSnaptr` if `reg.x.outboundProxy.address` is a hostname and `reg.x.outboundProxy.port` is 0 or Null, do NAPTR then SRV look-ups to try to discover the transport, ports and servers, as per RFC 3263. If `reg.x.outboundProxy.address` is an IP address, or a port is given, then UDP is used. **TCPpreferred** TCP is the preferred transport, UDP is used if TCP fails. **UDPOnly** only UDP will be used. **TLS** if TLS fails, transport fails. Leave port field empty (will default to `5061`) or set to `5061`. **TCPOnly** only TCP will be used.		
reg.x.proxyRequire	**string**	**Null**
The string that needs to be entered in the Proxy-Require header. If Null, no Proxy-Require will be sent.		
reg.x.ringType	**default, ringer1 to ringer24**	**ringer2**
The ringer to be used for calls received by this registration. The default is the first non-silent ringer. The configuration parameter `reg.x.ringtype` correctly uses the ringtones `ringer13` or `ringer14` only if `np.normal.ringing.calls.tonePattern` is set to `default` or if `reg.x.ringtype` is used by multiple line registrations. If you use the configuration parameters `ringer13` and `ringer14` on a single registered line, the phone plays `SystemRing.wav`. When setting `reg.x. ringType` to `ringer13` or to `ringer14.` Using the Web Configuration Utility, the correct ringtone is played using the override parameter `np.normal.ringing.calls.tonePattern=ringer13.`		
reg.x.ringType.privateLine	**default, ringer1 to ringer24**	**default**
The ringer to be used for calls received by a private line connected to Microsoft Lync Server 2010.		
reg.x.serverAutoDiscovery	**0 or 1**	**1**
Determines whether or not to discover the server address automatically. This parameter is used with Microsoft Lync Server 2010.		
reg.x.serverFeatureControl.cf[1]	**0 or 1**	**0**
If 0, server-based call forwarding is not enabled. If 1, server based call forwarding is enabled. This parameter overrides `voIpProt.SIP.serverFeatureControl.cf`.		
reg.x.serverFeatureControl.dnd[1]	**0 or 1**	**0**
If 0, server-based do-not-disturb (DND) is not enabled. If 1, server-based DND is enabled and the call server has control of DND. This parameter overrides `voIpProt.SIP.serverFeatureControl.dnd`.		
reg.x.serverFeatureControl.localProcessing.cf	**0 or 1**	**1**
If 0 and `reg.x.serverFeatureControl.cf` is set to 1, the phone will not perform local Call Forward behavior. If set to 1, the phone will perform local Call Forward behavior on all calls received. This parameter overrides `voIpProt.SIP.serverFeatureControl.localProcessing.cf`.		

Parameter	*Permitted Values*	*Default*
reg.x.serverFeatureControl.localProcessing.dnd	**0 or 1**	**1**
If 0 and `reg.x.serverFeatureControl.dnd` is set to 1, the phone will not perform local DND call behavior. If set to 1, the phone will perform local DND call behavior on all calls received. This parameter overrides `voIpProt.SIP.serverFeatureControl.localProcessing.dnd`.		
reg.x.serverFeatureControl.signalingMethod	**string**	**serviceMsForwardContact**
Controls the method used to perform call forwarding requests to the server.		
reg.x.server.y.registerRetry.maxTimeout		**180 seconds**
Set the maximum period of time in seconds that you want the phone to try registering with the server.		
reg.x.srtp.enable[1]	**0 or 1**	**1**
If 0, the registration always declines SRTP offers. If 1, the registration accepts SRTP offers.		
reg.x.srtp.offer[1]	**0 or 1**	**0**
If 1, the registration includes a secure media stream description along with the usual non-secure media description in the SDP of a SIP INVITE. This parameter applies to the registration initiating (offering) a phone call. If 0, no secure media stream is included in SDP of a SIP invite.		
reg.x.srtp.require[1]	**0 or 1**	**0**
If 0, secure media streams are not required. If 1, the registration is only allowed to use secure media streams. Any offered SIP INVITEs must include a secure media description in the SDP or the call will be rejected. For outgoing calls, only a secure media stream description is included in the SDP of the SIP INVITE, meaning that the non-secure media description is not included. If this parameter set to 1, `reg.x.srtp.offer` will also be set to 1, regardless of the value in the configuration file.		
reg.x.srtp.simplifiedBestEffort	**0 or 1**	**0**
If 0, no SRTP is supported. If 1, negotiation of SRTP compliant with Microsoft Session Description Protocol Version 2.0 Extensions is supported. This parameter overrides `sec.srtp.simplifiedBestEffort`.		
reg.x.strictLineSeize	**0 or 1**	**0**
If 1, the phone is forced to wait for 200 OK on registration x when receiving a TRYING notify. If set to 0, dial prompt is provided immediately when you attemp to seize a shared line without waiting for a successful OK from the call server. This parameter overrides `voIpProt.SIP.strictLineSeize` for registration x.		
reg.x.tcpFastFailover	**0 or 1**	**0**
If 1, failover occurs based on the values of `reg.x.server.y.retryMaxCount` and `voIpProt.server.x.retryTimeOut`. If 0, a full 32 second RFC compliant timeout is used.		
reg.x.telephony	**0 or 1**	**1**
If 0, telephony calls are not enabled on this registration (use this value if the registration is used with Microsoft Office Communications Server 2007 R2 or Microsoft Lync 2010. If 1, telephony calls are enabled on this registration.		
reg.x.thirdPartyName	**string address**	**Null**
This field must match the `reg.x.address` value of the registration which makes up the part of a bridged line appearance (BLA). It must be Null in all other cases.		

Parameter	Permitted Values	Default
reg.x.type	**private or shared**	**private**
If set to private, use standard call signaling. If set to shared, augment call signaling with call state subscriptions and notifications and use access control for outgoing calls.		

[1] Change causes phone to restart or reboot.

You can list multiple registration servers for fault tolerance. In the table Registration Server Parameters, you can list four servers by using y=1 to 4. If the `reg.x.server.y.address` is not null, all of the parameters in the following table will override the parameters specified in `voIpProt.server.*`. The server registration parameters are listed in the following table:

Registration Server Parameters

Parameter	Permitted Values	Default
reg.x.server.H323.y.expires	**positive integer**	**3600**
Desired registration period.		
reg.x.server.y.address	**dotted-decimal IP address or hostname**	**Null**
The IP address or host name of a SIP server that accepts registrations. If not Null, all of the parameters in this table will override the parameters specified in `voIpProt.server.*`. Notes: If this parameter is set, it will take precedence even if the DHCP server is available.		
reg.x.server.y.expires	**positive integer, minimum 10**	**3600**
The phone's requested registration period in seconds. Note: The period negotiated with the server may be different. The phone will attempt to re-register at the beginning of the overlap period. For example, if `expires="300"` and `overlap="5"`, the phone will re-register after 295 seconds (300–5).		
reg.x.server.y.expires.lineSeize	**0 to 65535**	**30**
Requested line-seize subscription period.		
reg.x.server.y.expires.overlap	**5 to 65535**	**60**
The number of seconds before the expiration time returned by server x at which the phone should try to re-register. The phone will try to re-register at half the expiration time returned by the server if the server value is less than the configured overlap value.		

Parameter	*Permitted Values*	*Default*
reg.x.server.y.failOver.failBack.mode	**newRequests** **DNSTTL** **registration** **duration**	**newRequests**
The mode for failover failback (this parameter overrides `voIpProt.server.x.failOver.failBack.mode`): `newRequests` – all new requests are forwarded first to the primary server regardless of the last used server. `DNSTTL` – the phone tries the primary server again after a timeout equal to the DNS TTL configured for the server that the phone is registered to. `registration` – the phone tries the primary server again when the registration renewal signaling begins. `duration` – the phone tries the primary server again after the time specified by `reg.x.server.y.failOver.failBack.timeout`.		
reg.x.server.y.failOver.failBack.timeout	**0, 60 to 65535**	**3600**
The time to wait (in seconds) before failback occurs (overrides `voIpProt.server.x.failOver.failBack.timeout`).If the fail back mode is set to Duration, the phone waits this long after connecting to the current working server before selecting the primary server again. If 0, the phone will not fail-back until a fail-over event occurs with the current server.		
reg.x.server.y.failOver.failRegistrationOn	**0 or 1**	**0**
When set to 1, and the reRegisterOn parameter is enabled, the phone will silently invalidate an existing registration (if it exists), at the point of failing over. When set to 0, and the reRegisterOn parameter is enabled, existing registrations will remain active. This means that the phone will attempt failback without first attempting to register with the primary server to determine if it has recovered.		
reg.x.server.y.failOver.onlySignalWithRegistered	**0 or 1**	**1**
When set to 1, and the reRegisterOn and failRegistrationOn parameters are enabled, no signaling is accepted from or sent to a server that has failed until failback is attempted or failover occurs. If the phone attempts to send signaling associated with an existing call via an unregistered server (for example, to resume or hold a call), the call will end. No SIP messages will be sent to the unregistered server. When set to 0, and the reRegisterOn and failRegistrationOn parameters are enabled, signaling will be accepted from and sent to a server that has failed (even though failback hasn't been attempted or failover hasn't occurred).		
reg.x.server.y.failOver.reRegisterOn	**0 or 1**	**0**
This parameter overrides the `voIpProt.server.x.failOver.reRegisterOn`. When set to 1, the phone will attempt to register with (or via, for the outbound proxy scenario), the secondary server. If the registration succeeds (a 200 OK response with valid expires), signaling will proceed with the secondary server. When set to 0, the phone won't attempt to register with the secondary server, since the phone will assume that the primary and secondary servers share registration information.		
reg.x.server.y.lcs	**0 or 1**	**0**
If 0, the Microsoft Live Communications Server (LSC) is not supported. If 1, LCS is supported for registration x.		
reg.x.server.y.useOutboundProxy	**0 or 1**	**1**
Specify whether or not to use the outbound proxy specified in `reg.x.outboundProxy.address` for server x. This parameter overrides `voIpProt.server.x.useOutboundProxy` for registration x.		

Parameter	*Permitted Values*	*Default*
reg.x.server.y.port	**0, 1 to 65535**	**Null**
The port of the sip server that specifies registrations. If 0, the port used depends on `reg.x.server.y.transport`.		
reg.x.server.y.register	**0 or 1**	**1**
If 0, calls can be routed to an outbound proxy without registration. See voIpProt.server.x.register. For more information, see Technical Bulletin 5844: SIP Server Fallback Enhancements on Polycom Phones.		
reg.x.server.y.registerRetry.baseTimeOut	**10 - 120**	**60**
The base time period to wait before a registration retry. Used in conjunction with `reg.x.server.y.registerRetry.maxTimeOut` to determine how long to wait. The algorithm is defined in RFC 5626.		
reg.x.server.y.registerRetry.maxTimeOut	**60 - 1800**	**60**
The maximum time period to wait before a registration retry. Used in conjunction with `reg.x.server.y.registerRetry.baseTimeOut` to determine how long to wait. The algorithm is defined in RFC 5626.		
reg.x.server.y.retryMaxCount	**0 to 20**	**3**
If set to 0, 3 is used. The number of retries that will be attempted before moving to the next available server.		
reg.x.server.y.retryTimeOut	**0 to 65535**	**0**
The amount of time (in milliseconds) to wait between retries. If 0, use standard RFC 3261 signaling retry behavior.		
reg.x.server.y.specialInterop	**standard, ocs2007r2, lcs2005, lync2010**	**standard**
Specify if this registration should support Microsoft Office Communications Server 2007 R2 (ocs2007r2), Microsoft Live Communications Server 2005 (lcs2005), or Microsoft Lync 2010 (lync2010).		
reg.x.server.y.transport	**DNSnaptr, TCPpreferred, UDPOnly, TLS, TCPOnly**	**DNSnaptr**
The transport method the phone uses to communicate with the SIP server. `Null` or `DNSnaptr` – if `reg.x.server.y.address` is a hostname and `reg.x.server.y.port` is 0 or Null, do NAPTR then SRV look-ups to try to discover the transport, ports and servers, as per RFC 3263. If `reg.x.server.y.address` is an IP address, or a port is given, then UDP is used. `TCPpreferred` – TCP is the preferred transport; UDP is used if TCP fails. `UDPOnly` – only UDP will be used. `TLS` – if TLS fails, transport fails. Leave port field empty (will default to `5061`) or set to `5061`. `TCPOnly` – only TCP will be used.		

<request/>

The parameters listed in the table Configuration Request Parameter control the phone’s behavior when a request for restart or reconfiguration is received.

Configuration Request Parameter

Parameter	*Permitted Values*	*Default*
request.delay.type[1]	**audio, call**	**call**
Specify when the phone should process a request for a restart or reconfiguration. If set to `audio`, the request will be executed once there is no active audio on the phone—regardless of the call state. If set to `call`, the request should be executed once there are no calls —in any state—on the phone.		

[1] Change causes phone to restart or reboot.

<roaming_buddies/>

The parameters listed in the table Roaming Buddies Parameters is used in conjunction with Microsoft Lync on most Polycom phones.

Roaming Buddies Parameters

Parameter	*Permitted Value*	*Default*
roaming_buddies.reg	**1 to 34**	**Null**
The index of the registration which has roaming buddies support enabled. If Null, the roaming buddies feature is disabled. **Note**: This parameter must be set if the call server is Microsoft Lync.		

<roaming_privacy/>

The parameters in the table Roaming Privacy Parameters are used conjunction with Microsoft Lync Server on Lync-enabled Polycom phones.

Roaming Privacy Parameters

Parameter	*Permitted Value*	*Default*
roaming_privacy.reg	**1 to 34**	**Null**
Specify the index of the registration/line that has roaming privacy support enabled. If Null, roaming privacy is disabled.		

<saf/>

The phone uses built-in wave files for some sound effects. The built-in wave files can be replaced with files downloaded from the provisioning server or from the Internet. However, these are stored in volatile memory so the files will need to remain accessible should the phone need to be rebooted. Files will be truncated to a maximum size of 300 kilobytes.

The following sampled audio WAVE (.wav) file formats are supported:

- mono 8 kHz G.711 u-Law
- G.711 A-Law
- L16/16000 (16-bit, 16 kHz sampling rate, mono)
- L16/32000 (16-bit, 32 kHz sampling rate, mono)
- L16/48000 (16-bit, 48 kHz sampling rate, mono)

In the table Sampled Audio File Parameters, *x* is the sampled audio file number.

Sampled Audio File Parameters

Parameter	*Permitted Values*	*Default*
saf.x	**Null or valid path name or an RFC 1738-compliant URL to a HTTP, FTP, or TFTP wave file resource.**	
If Null, the phone will use a built-in file. If set to a path name, the phone will attempt to download this file at boot time from the provisioning server. If set to a URL, the phone will attempt to download this file at boot time from the Internet. Note: A TFTP URL is expected to be in the format: tftp://<host>/[pathname]<filename>, for example: tftp://somehost.example.com/sounds/example.wav . Note: See the above wave file format restrictions.		

The table Default Sample Audio File Usage defines the default usage of the sampled audio files with the phone:

Default Sample Audio File Usage

Sampled Audio File Number	*Default Use (Pattern Reference)*
1	Ringer 12 (se.pat.misc.welcome)
2	Ringer 13 (se.pat.ringer.ringer15)
3	Ringer 14 (se.pat.ringer.ringer16)
4	Ringer 15 (se.pat.ringer.ringer17)
5	Ringer 16 (se.pat.ringer.ringer18)
6	Ringer 17 (se.pat.ringer.ringer19)
7	Ringer 18 (se.pat.ringer.ringer20)

Sampled Audio File Number	*Default Use (Pattern Reference)*
8	Ringer 19 (`se.pat.ringer.ringer21`)
9	Ringer 20 (`se.pat.ringer.ringer22`)
10	Ringer 21 (`se.pat.ringer.ringer23`)
11	Ringer 22 (`se.pat.ringer.ringer24`)
12 to 24	Not Used

<se/>

The table Sound Effect Parameters lists configurable sound effect parameters. You can also configure sound effect patterns in <pat/> and ringtones in <rt/>. The phone uses both synthesized (based on the chord-sets, see <chord/>) and sampled audio sound effects. Sound effects are defined by patterns: rudimentary sequences of chord-sets, silence periods, and wave files.

Sound Effect Parameters

Parameter	*Permitted Values*	*Default*
se.appLocalEnabled[1]	**0 or 1**	**1**
If set to 1, local user interface sound effects such as confirmation/error tones, will be enabled.		
se.destination	**chassis, headset, handset, active**	**1**
The transducer or audio device that plays sound effects and alerts. Choose from the `chassis` (speakerphone), `headset` (if connected), `handset`, or the `active` destination. If `active`, alerts will play from the destination that is currently in use. For example, if you are in a call on the handset, a new incoming call will ring on the handset.		
se.stutterOnVoiceMail	**0 or 1**	**1**
If set to 1, a stuttered dial tone is used in place of a normal dial tone to indicate that one or more voicemail messages are waiting at the message center.		

<pat/>

Patterns use a simple script language that allows different chord sets or wave files to be strung together with periods of silence. The script language uses the instructions shown in the table Sound Effects Pattern Types.

Sound Effects Pattern Types

Instruction	*Meaning*
sampled (n)	**Play sampled audio file n**
Example: `se.pat.misc.SAMPLED_1.inst.1.type =”sampled”` (sampled audio file instruction type) `se.pat.misc.SAMPLED_1.inst.1.value =”2”` (specifies sampled audio file 2)	
chord (n, d)	**Play chord set n (d is optional and allows the chord set ON duration to be overridden to d milliseconds)**
Example: `se.pat.callProg.busyTone.inst.2.type = “chord”` (chord set instruction type) `se.pat.callProg.busyTone.inst.2.value = “busyTone”` (specifies sampled audio file busyTone) `se.pat.callProg.busyTone.inst.2.param = “2000”` (override ON duration of chord set to 2000 milliseconds)	
silence (d)	**Play silence for d milliseconds (Rx audio is not muted)**
Example: `se.pat.callProg.bargeIn.inst.3.type = “silence”` (silence instruction type) `se.pat.callProg.bargeIn.inst.3.value = “300”` (specifies silence is to last 300 milliseconds)	
branch (n)	**Advance n instructions and execute that instruction (n must be negative and must not branch beyond the first instruction)**
Example: `se.pat.callProg.alerting.inst.4.type = “branch”` (branch instruction type) `se.pat.callProg.alerting.inst.4.value = “-2”` (step back 2 instructions and execute that instruction)	

In the table Sound Effects Pattern Parameters, *x* is the pattern name, *y* is the instruction number. Both *x* and *y* need to be sequential. There are three categories cat of sound effect patterns: `callProg` (Call Progress Patterns), `ringer` (Ringer Patterns) and `misc` (Miscellaneous Patterns).

Sound Effects Pattern Parameters

Parameter	*Permitted Values*
se.pat.cat.x.name	**UTF-8 encoded string**
Sound effects name, where cat is `callProg`, `ringer`, or `misc`.	

Parameter	*Permitted Values*
se.pat.cat.x.inst.y.type	**sampled, chord, silence, branch**
Type of sound effect, where cat is `callProg`, `ringer`, or `misc`.	
se.pat.cat.x.inst.y.value	**String**
The instruction: `sampled` – sampled audio file number, `chord` – type of sound effect, `silence` – silence duration in ms, `branch` – number of instructions to advance. `cat` is `callProg`, `ringer`, or `misc`.	

The table Call Progress Tone Pattern Names shows the call progress pattern names and their descriptions:

Call Progress Tone Pattern Names

Call Progress Pattern Name	*Description*
alerting	Alerting
bargeIn	Barge-in tone
busyTone	Busy tone
callWaiting	Call waiting tone
callWaitingLong	Call waiting tone long (distinctive)
confirmation	Confirmation tone
dialTone	Dial tone
howler	Howler tone (off-hook warning)
intercom	Intercom announcement tone
msgWaiting	Message waiting tone
precedenceCallWaiting	Precedence call waiting tone
precedenceRingback	Precedence ringback tone
preemption	Preemption tone
precedence	Precedence tone
recWarning	Record warning
reorder	Reorder tone
ringback	Ringback tone
secondaryDialTone	Secondary dial tone
stutter	Stuttered dial tone

The table Ringtone Pattern Names shows the ring pattern names and their default descriptions:

Ringtone Pattern Names

Parameter Name	*Ringtone Name*	*Description*
ringer1	Silent Ring	Silent ring
ringer2	Low Trill	Long single A3 Db3 major warble
ringer3	Low Double Trill	Short double A3 Db3 major warble
ringer4	Medium Trill	Long single C3 E3 major warble
ringer5	Medium Double Trill	Short double C3 E3 major warble
ringer6	High Trill	Long single warble 1
ringer7	High Double Trill	Short double warble 1
ringer8	Highest Trill	Long single Gb3 A4 major warble
ringer9	Highest Double Trill	Short double Gb3 A4 major warble
ringer10	Beeble	Short double E3 major
ringer11	Triplet	Short triple C3 E3 G3 major ramp
ringer12	Ringback-style	Short double ringback
ringer13	Low Trill Precedence	Long single A3 Db3 major warble Precedence
ringer14	Ring Splash	Splash
ringer15	Ring16	Sampled audio file 1
ringer16	Ring17	Sampled audio file 2
ringer17	Ring18	Sampled audio file 3
ringer18	Ring19	Sampled audio file 4
ringer19	Ring20	Sampled audio file 5
ringer20	Ring21	Sampled audio file 6
ringer21	Ring22	Sampled audio file 7
ringer22	Ring23	Sampled audio file 8
ringer23	Ring24	Sampled audio file 9
ringer24	Ring25	Sampled audio file 10

Note: Silent Ring

Silent ring will provide a visual indication of an incoming call, but no audio indication. Sampled audio files 1 to 10 all use the same built-in file unless that file has been replaced with a downloaded file. For more information, see <saf/>.

The table Miscellaneous Pattern Names shows the miscellaneous patterns and their descriptions:

Miscellaneous Pattern Names

Miscellaneous pattern name	*Description*
instant message	New instant message
local hold notification	Local hold notification
message waiting	New message waiting indication
negative confirmation	Negative confirmation
positive confirmation	Positive confirmation
remote hold notification	Remote hold notification
welcome	Welcome (boot up)

<rt/>

Ringtone is used to define a simple class of ring to be applied based on some credentials that are usually carried within the network protocol. The ring class includes parameters such as call-waiting and ringer index, if appropriate. The ring class can use one of four types of ring that are defined as follows:

- **ring** Play a specified ring pattern or call waiting indication
- **visual** Provide only a visual indication (no audio) of an incoming call, no ringer needs to be specified
- **answer** Provide auto-answer on an incoming call
- **ring-answer** Provide auto-answer on an incoming call after a certain number of rings

Note: Using the Answer Ring Type

The auto-answer on incoming call is currently only applied if there is no other call in progress on the phone at the time.

The phone supports the following ring classes: **default**, **visual**, **answerMute**, **autoAnswer**, **ringAnswerMute**, **ringAutoAnswer**, **internal**, **external**, **emergency**, **precedence**, **splash**, and **custom<*y*>** where y is 1 to 17.

In the following table, x is the ring class name.

Caution: Ringtone Parameters Will Not Work After a Software Downgrade

If a phone has been upgraded to Polycom UC Software 4.0.0 and then downgraded to SIP 3.2.3 or earlier, the ringtone parameters will be unusable due to configuration parameters name changes in UC Software 4.0.0.

Sound Effects Ringtone Parameters

Parameter	*Permitted Values*
se.rt.enabled	**0 or 1 (default)**
If **0**, the ringtone feature is not enabled on the phone. If **1** (default), the ringtone feature is enabled.	
se.rt.modification.enabled	**0 or 1 (default)**
A flag to determine whether or not to allow user modification (through phone's user interface) of the pre-defined ringtone enabled for modification.	
se.rt.<ringClass>.callWait	**callWaiting, callWaitingLong, precedenceCallWaiting**
The call waiting tone to be used for this class of ring. The call waiting should match one defined in Call Progress Tone Pattern Names. The default call waiting tone is `callWaiting`.	
se.rt.<ringClass>.name	**UTF-8 encoded string**
The answer mode for a ringtone. Used for identification purposes in the user interface.	
se.rt.<ringClass>.ringer	**default, ringer1 to ringer24**
The ringtone to be used for this class of ring. The ringer should match one of Ringtone Pattern Names. The default ringer is `ringer2` .	
se.rt.<ringClass>.timeout	**1 to 60000 only relevant if the type is set to ring-answer**
The duration of the ring in milliseconds before the call is auto answered. The default is 2000.	
se.rt.<ringClass>.type	**ring, visual, answer, ring-answer**
The answer mode for a ringtone as defined in list earlier in this section.	

<sec/>

The parameters listed in the table General Security Parameters affects the security features of the phone. The configuration parameter is defined as follows:

General Security Parameters

Parameter	*Permitted Values*	*Default*
sec.tagSerialNo[1]	**0 or 1**	**0**
If 0, the phone does not advertise its serial number (MAC address) through protocol signaling. If 1, the phone may advertise its serial number through protocol signaling.		

[1] Change causes phone to restart or reboot.

This parameter also includes:

- <encryption/>
- <pwd/><length/>
- <srtp/>
- <dot1x><eapollogoff/>
-
- <hostmovedetect/>
- <TLS/>
  - <profile/>
  - <profileSelection/>

<encryption/>

The table File Encryption Parameters lists available encryption parameters.

File Encryption Parameters

Parameter	*Permitted Values*	*Default*
sec.encryption.upload.callLists[1]	**0 or 1**	**0**
The encryption on the phone-specific call lists that is uploaded to the provisioning server. If 0, the call list is uploaded unencrypted regardless of how it was downloaded, the directory replaces whatever phone-specific call list is on the server, even if the file on the server is encrypted. If 1, the call list is uploaded encrypted regardless of how it was downloaded. The file replaces any existing phone-specific call lists file on the server.		

Parameter	*Permitted Values*	*Default*
sec.encryption.upload.config	**0 or 1**	**0**
The encryption on the phone-specific configuration file created and uploaded to the provisioning server when the user selects **Upload Configuration** from the phone menu. If 0, the file is uploaded unencrypted, and overwrites whatever phone-specific configuration file is on the server, even if the file on the server is encrypted. If 1, the file is uploaded encrypted and replaces any existing phone-specific configuration file on the server. If there is no encryption key on the phone, the file is not uploaded.		
sec.encryption.upload.dir[1]	**0 or 1**	**0**
The encryption on the phone-specific contact directory that is uploaded to the provisioning server. If 0, the directory is uploaded unencrypted regardless of how it was downloaded, the directory replaces whatever phone-specific contact directory is on the server, even if the file on the server is encrypted. If 1, the directory is uploaded encrypted regardless of how it was downloaded. The file replaces any existing phone-specific contact directory file on the server.		
sec.encryption.upload.overrides	**0 or 1**	**0**
The encryption on the phone-specific **<MACaddress>-phone.cfg** override file that is uploaded to the server. If 0, the file is uploaded unencrypted regardless of how it was downloaded, the file replaces whatever file was on the server, even if the file on the server is encrypted. If 1, the file is uploaded encrypted regardless of how it was downloaded. The file replaces any existing phone-specific override file on the server.		

[1] Change causes phone to restart or reboot.

<pwd/><length/>

The table Password Length Parameters lists configurable password length parameters.

Password Length Parameters

Parameter	*Permitted Values*	*Default*
sec.pwd.length.admin[1]	**0-32**	**1**
The minimum length for administrator passwords changed using the phone. Use 0 to allow null passwords.		
sec.pwd.length.user[1]	**0-32**	**2**
The minimum length for user passwords changed using the phone. Use 0 to allow null passwords.		

[1] Change causes phone to restart or reboot.

<srtp/>

As per RFC 3711, you cannot turn off authentication of RTCP. The table SRTP Parameters lists SRTP parameters.

SRTP Parameters

Parameter	*Permitted values*	*Defaults*
sec.srtp.answerWithNewKey	**0 or 1**	**1**
If 0, a new key is not provided when answering a call. If 1, a new key is provided when answering a call.		
sec.srtp.enable[1]	**0 or 1**	**1**
If 0, the phone always declines SRTP offers. If 1, the phone accepts SRTP offers. Note: The defaults for SIP 3.2.0 was 0 when Null or not defined.		
sec.srtp.key.lifetime[1]	**0, positive integer minimum 1024 or power of 2 notation**	**Null**
The lifetime of the master key used for the cryptographic parameter in SDP. The value specified is the number of SRTP packets. If 0, the master key lifetime is not set. If set to a valid value (at least 1024, or a power such as 2^10), the master key lifetime is set. When the lifetime is set, a re-invite with a new key will be sent when the number or SRTP packets sent for an outgoing call exceeds half the value of the master key lifetime. Note: Setting this parameter to a non-zero value may affect the performance of the phone.		
sec.srtp.mki.enabled[1]	**0 or 1**	**0**
The master key identifier (MKI) is an optional parameter for the cryptographic parameter in the SDP that uniquely identifies the SRTP stream within an SRTP session. MKI is expressed as a pair of decimal numbers in the form: `|mki:mki_length|` where `mki` is the MKI value and `mki_length` its length in bytes. If 1, a four-byte MKI parameter is sent within the SDP message of the SIP INVITE / 200 OK. If 0, the MKI parameter is not sent.		
sec.srtp.mki.length[1]	**1 to 4**	**4**
The length of the master key identifier (MKI), in bytes. Microsoft Lync offers 1-byte MKIs.		
sec.srtp.mki.startSessionAtOne	**0 or 1**	**0**
If set to 1, use an MKI value of 1 at the start of an SDP session. If set to 0, the MKI value will increment for each new crypto key.		
sec.srtp.offer[1]	**0 or 1**	**0**
If 1, the phone includes a secure media stream description along with the usual non-secure media description in the SDP of a SIP INVITE. This parameters applies to the phone initiating (offering) a phone call. If 0, no secure media stream is included in SDP of a SIP invite.		
sec.srtp.offer.HMAC_SHA1_32[1]	**0 or 1**	**0**
If 1, a crypto line with the `AES_CM_128_HMAC_SHA1_32` crypto-suite will be included in offered SDP. If 0, the crypto line is not included.		
sec.srtp.offer.HMAC_SHA1_80[1]	**0 or 1**	**1**
If 1, a crypto line with the `AES_CM_128_HMAC_SHA1_80` crypto-suite will be included in offered SDP. If 0, the crypto line is not included.		

Parameter	*Permitted values*	*Defaults*
sec.srtp.padRtpToFourByteAlignment[1]	**0 or 1**	**0**
Packet padding may be required when sending or receiving video from other video products. If 1, RTP packet padding is needed. If 0, no packet padding is needed.		
sec.srtp.require[1]	**0 or 1**	**0**
If 0, secure media streams are not required. If 1, the phone is only allowed to use secure media streams. Any offered SIP INVITEs must include a secure media description in the SDP or the call will be rejected. For outgoing calls, only a secure media stream description is included in the SDP of the SIP INVITE, meaning that the non-secure media description is not included. If this parameter set to 1, `sec.srtp.offer` will also be set to 1, regardless of the value in the configuration file.		
sec.srtp.requireMatchingTag[1]	**0 or 1**	**1**
If 0, the tag values in the crypto parameter in an SDP answer are ignored. If 1, the tag values must match.		
sec.srtp.sessionParams.noAuth.offer[1]	**0 or 1**	**0**
If 0, authentication of RTP is offered. If 1, no authentication of RTP is offered; a session description that includes the `UNAUTHENTICATED_SRTP` session parameter is sent when initiating a call.		
sec.srtp.sessionParams.noAuth.require[1]	**0 or 1**	**0**
If 0, authentication of RTP is required. If 1, no authentication of RTP is required; a call placed to a phone configured with this parameter must offer the UNAUTHENTICATED_SRTP session parameter in its SDP. If this parameter is set to 1, `sec.srtp.sessionParams.noAuth.offer` will also be set to 1, regardless of the value in the configuration file.		
sec.srtp.sessionParams.noEncrypRTCP.offer[1]	**0 or 1**	**0**
If 0, encryption of RTCP is offered. If 1, no encryption of RTCP is offered; a session description that includes the UNENCRYPTED_SRTCP session parameter is sent when initiating a call.		
sec.srtp.sessionParams.noEncrypRTCP.require[1]	**0 or 1**	**0**
If set to 0, encryption of RTCP is required. If set to 1, no encryption of RTCP is required; a call placed to a phone configured with `noAuth.require` must offer the UNENCRYPTED_SRTCP session parameter in its SDP. If this parameter is set to 1, `sec.srtp.sessionParams.noEncryptRTCP.offer` will also be set to 1, regardless of the value in the configuration file.		
sec.srtp.sessionParams.noEncrypRTP.offer[1]	**0 or 1**	**0**
If 0, encryption of RTP is offered. If 1, no encryption of RTP is offered; a session description that includes the UNENCRYPTED_SRTP session parameter is sent when initiating a call.		
sec.srtp.sessionParams.noEncrypRTP.require[1]	**0 or 1**	**0**
If 0, encryption of RTP is required. If 1, no encryption of RTP is required. A call placed to a phone configured with noAuth.require must offer the UNENCRYPTED_SRTP session parameter in its SDP. If set to 1, sec.srtp.sessionParams.noEncryptRTP.offer will also be set to 1, regardless of the value in the configuration file.		
sec.srtp.simplifiedBestEffort	**0 or 1**	**0**
If 0, no SRTP is supported. If 1, negotiation of SRTP compliant with Microsoft Session Description Protocol Version 2.0 Extensions is supported.		

[1] Change causes phone to restart or reboot.

<dot1x><eapollogoff/>

The table 802.1X EAP over LAN (EAPOL) Logoff Parameters lists configurable parameters.

802.1X EAP over LAN (EAPOL) Logoff Parameters

Parameter	*Permitted Values*	*Default*
sec.dot1x.eapollogoff.enabled[1]	**0 or 1**	**0**
If 0, the phone will not send an EAPOL Logoff message on behalf of the disconnected supplicant. If 1, the feature is enabled and the phone will send an EAPOL Logoff message on behalf of the disconnected supplicant connected to the phone’s secondary (PC) port.		
sec.dot1x.eapollogoff.lanlinkreset[1]	**0 or 1**	**0**
If 0, the phone software will not reset (recycle) the LAN port link in the application initiation stage. If 1, the LAN port link will be reset in the application initiation stage.		

[1] Change causes phone to restart or reboot.

<hostmovedetect/>

The table Host Movement Detection Parameters lists configurable parameters.

Host Movement Detection Parameters

Parameter	*Permitted Values*	*Default*
sec.hostmovedetect.cdp.enabled[1]	**0 or 1**	**0**
If set to 1, the phone software will unconditionally send a CDP packet (to the authenticator switch port) to indicate a host has been connected or disconnected to its secondary (PC) port.		
sec.hostmovedetect.cdp.sleepTime[1]	**0 to 60000**	**1000**
If `sec.hostmovedetect.cdp.enabled` is set to 1, then there will be an x microsecond time interval between two consecutive link–up state change reports. This will reduce the frequency of dispatching CDP packets.		

[1] Change causes phone to restart or reboot.

<TLS/>

The table TLS Parameters lists configurable TLS parameters. For the list of configurable ciphers, see Configurable TLS Cipher Suites.

This parameter also includes <profile/>and <profileSelection/>.

TLS Parameters

Parameter	*Permitted Values*	*Default*
sec.TLS.browser.cipherList	**String**	**NoCipher**
The cipher list for browser. The format for the cipher list uses openSSL syntax found here: OpenSSL Ciphers.		
sec.TLS.cipherList	**String**	**"RSA:!EXP:!LOW:!NULL:!MD5:@STRENGTH"**
The global cipher list parameter. The format for the cipher list uses openSSL syntax found here: OpenSSL Ciphers.		
sec.TLS.customCaCert.x	**String**	**Null**
The custom certificate for TLS Application Profile x (x= 1 to 6).		
sec.TLS.customDeviceCert.x	**String**	**Null**
The custom device certificate for TLS Application Profile x (x= 1 to 6).		
sec.TLS.customDeviceKey.x	**String**	**Null**
The custom device certificate private key for TLS Application Profile x (x= 1 to 6).		
sec.TLS.LDAP.cipherList	**String**	**NoCipher**
The cipher list for the corporate directory. The format for the cipher list uses openSSL syntax found here: OpenSSL Ciphers.		
sec.TLS.prov.cipherList	**String**	**NoCipher**
The cipher list for provisioning. The format for the cipher list uses openSSL syntax found here: OpenSSL Ciphers.		
sec.TLS.SIP.cipherList	**String**	**NoCipher**
The cipher list for SIP. The format for the cipher list uses openSSL syntax found here: OpenSSL Ciphers.		
sec.TLS.SIP.strictCertCommonNameValidation	**0 or 1**	**1**
If 1, enable common name validation for SIP.		
sec.TLS.syslog.cipherList	**String**	**NoCipher**
The cipher list for syslog. The format for the cipher list uses openSSL syntax found here: OpenSSL Ciphers.		
sec.TLS.xmpp.cipherList	**String**	**NoCipher**
The cipher list for CMA presence. The format for the cipher list uses openSSL syntax found here: OpenSSL Ciphers.		

<profile/>

Profiles are a collection of related security parameters. The table TLS Profile Parameters lists TLS profile parameters. There are two platform profiles and six application profiles.

TLS Profile Parameters

Parameter	*Permitted Values*	*Default*
sec.TLS.profile.x.caCert.application1 **Application CA 1**	**0 or 1**	**1**
sec.TLS.profile.x.caCert.application2 **Application CA 2**	**0 or 1**	**1**
sec.TLS.profile.x.caCert.application3 **Application CA 3**	**0 or 1**	**1**
sec.TLS.profile.x.caCert.application4 **Application CA 4**	**0 or 1**	**1**
sec.TLS.profile.x.caCert.application5 **Application CA 5**	**0 or 1**	**1**
sec.TLS.profile.x.caCert.application6 **Application CA 6**	**0 or 1**	**1**
sec.TLS.profile.x.caCert.platform1 **Platform CA 1**	**0 or 1**	**1**
sec.TLS.profile.x.caCert.platform2 **Platform CA 2**	**0 or 1**	**1**
Specify which CA certificates should be used for TLS Application Profile x (where x is 1 to 6). If set to 0, the CA will not be used. If set to 1, the CA will be used.		
sec.TLS.profile.x.caCert.defaultList	**String**	**Null**
The list of default CA certificates for TLS Application Profile x (x= 1 to 6).		
sec.TLS.profile.x.cipherSuite	**String**	**Null**
The cipher suite for TLS Application Profile x (where x is 1 to 6).		
sec.TLS.profile.x.cipherSuiteDefault	**0 or 1**	**1**
If 0, use the custom cipher suite for TLS Application Profile x (x= 1 to 6). If 1, use the default cipher suite.		
sec.TLS.profile.x.deviceCert	**Polycom, Platform1, Platform2, Application1, Application2, Application3, Application4, Application5, Application6**	**Polycom**
The device certificate to use for TLS Application Profile x (x = 1 to 6).		

<profileSelection/>

You can configure the parameters listed in the table TLS Profile Selection Parameters to choose the platform profile or application profile to use for each TLS application.

The permitted values are:

- PlatformProfile1
- PlatformProfile2
- ApplicationProfile1
- ApplicationProfile2
- ApplicationProfile3
- ApplicationProfile4
- ApplicationProfile5
- ApplicationProfile6

TLS Profile Selection Parameters

Parameter	*Permitted Values*	*Default*
sec.TLS.profileSelection.browser	**a TLS profile**	**PlatformProfile1**
The TLS platform profile or TLS application profile (see preceding list) to use for the browser or microbrowser.		
sec.TLS.profileSelection.LDAP	**a TLS profile**	**PlatformProfile1**
The TLS platform profile or TLS application profile (see preceding list) to use for the Corporate Directory.		
sec.TLS.profileSelection.SIP	**a TLS profile**	**PlatformProfile1**
The TLS platform profile or TLS application profile (see preceding list) to use for SIP operations.		
sec.TLS.profileSelection.syslog	**PlatformProfile1 or PlatformProfile2**	**PlatformProfile1**
The TLS platform profile to use for syslog operations.		
sec.TLS.profileSelection.XMPP	**a TLS profile**	**PlatformProfile1**
The TLS platform profile or TLS application profile (see preceding list) to use for the CMA Directory.		

<softkey/>

The table Soft Key Customization Parameters lists parameters you can use to customize soft keys on the phone interface. Note that `feature.enhancedFeatureKeys.enabled` must be enabled (set to 1) to use the Configurable Soft Key feature.

The configuration parameter is defined as follows (where x=1 to a maximum number of 10 soft keys).

Soft Key Customization Parameters

Parameter	*Permitted Values*	*Default*
softkey.feature.basicCallManagement.redundant	**0 or 1**	**1**
Control the display of the **Hold**, **Transfer**, and **Conference** soft keys. If set to 0 and the phone has hard keys mapped for **Hold**, **Transfer**, and **Conference** functions (all must be mapped), none of the soft keys are displayed. If set to 1, all of these soft keys are displayed.		
softkey.feature.buddies	**0 or 1**	**1**
If 0, the **Buddies** soft key is not displayed. If 1, the soft key is displayed (if `pres.idleSoftKeys` is set to 1).		
softkey.feature.callers	**0 or 1**	**0**
If 1, the **Callers** soft key displays on all platforms. If 0, the **Callers** soft key is disabled for all platforms. The default value is 0.		
softkey.feature.directories	**0 or 1**	**0**
If 1, the **Directory** soft key displays on all platforms. If 0, the **Directory** soft key is disabled for all platforms. The default value is 0.		
softkey.feature.endcall	**0 or 1**	**1**
If 0, the **End Call** soft key is not displayed. If 1, the soft key is displayed.		
softkey.feature.forward	**0 or 1**	**1**
If 0, the **Forward** soft key is not displayed. If 1, the soft key is displayed.		
softkey.feature.join	**0 or 1**	**1**
Join two individual calls to form a conference. If 0, the **Join** soft key is not displayed. If 1, the soft key is displayed.		
softkey.feature.mystatus	**0 or 1**	**1**
If 0, the **MyStatus** soft key is not displayed. If 1, the soft key is displayed (if `pres.idleSoftKeys` is set to 1).		
softkey.feature.newcall	**0 or 1**	**1**
If 0, the **New Call** soft key is not displayed when there is an alternative way to place a call. If 1, the **New Call** soft key is displayed.		
softkey.feature.simplifiedSignIn	**0 or 1**	**0**
If 0, the **SignIn** soft key is not displayed. If 1 and `voIpProt.server.x.specialInterop` is `lync2010`, the **SignIn** soft key is displayed.		
softkey.feature.split	**0 or 1**	**1**
Split up a conference into individual calls. If 0, the **Split** soft key is not displayed. If 1, the soft key is displayed.		
softkey.x.action	**macro action string, 256 characters**	**Null**
The action or function for custom soft key x. This value uses the same macro action string syntax as an Enhanced Feature Key. For a list of actions, see Understanding Macro Definitions.		

Parameter	*Permitted Values*	*Default*
softkey.x.enable	**0 or 1**	**0**
If 0, the soft key x is disabled. If 1, the soft key is enabled.		
softkey.x.insert	**0 to 10**	**0**
The position on the phone screen for soft key x. For example, if the value is 3, the soft key will be displayed on the screen in the third position from the left. Note: If `softkey.x.precede` is configured, this value is ignored. If the insert location is greater than the number of soft keys, the key will be positioned last, after the other soft keys.		
softkey.x.label	**string**	**Null**
The text displayed on the soft key label. If Null, the label is determined as follows: If the soft key performs an Enhanced Feature Key macro action, the label of the macro will be used. If the soft key calls a speed dial, the label of the speed dial contact will be used. If the soft key performs chained actions, the label of the first action is used. If the soft key label is Null and none of the preceding criteria are matched, the label will be blank.		
softkey.x.precede	**0 or 1**	**0**
If 0, soft key x is positioned in the first empty space from the left. If 1, the soft key is displayed before (to the left of) the first default soft key.		
softkey.x.use.active **Display in the active call state**	**0 or 1**	**0**
softkey.x.use.alerting **Display in the alerting state**	**0 or 1**	**0**
softkey.x.use.dialtone **Display in the dial tone state**	**0 or 1**	**0**
softkey.x.use.hold **Display in the hold state**	**0 or 1**	**0**
softkey.x.use.idle **Display in the idle state**	**0 or 1**	**0**
softkey.x.use.proceeding **Display in the proceeding state**	**0 or 1**	**0**
softkey.x.use.setup **Display in the proceeding state**	**0 or 1**	**0**
If 0, the soft key is not displayed when the phone is in the call state. If 1, the soft key is displayed when the phone is in the call state.		

<tcpIpApp/>

This parameter includes:

- <dhcp/>
- <dns/>
- <ice/>
- <sntp/>

- <port/><rtp/>
- <keepalive/>
- <fileTransfer/>

<dhcp/>

The DHCP parameters listed in the table DHCP Parameters enable you to change how the phone reacts to DHCP changes.

DHCP Parameters

Parameter	*Permitted Values*	*Default*
tcpIpApp.dhcp.releaseOnLinkRecovery	**0 or 1**	**1**
If 0, no DHCP release occurs. If 1, a DHCP release is performed after the loss and recovery of the network.		

<dns/>

The <dns/> parameters listed in the table Domain Name System (DNS) Parameters enables you to set Domain Name System (DNS). However, any values set through DHCP will have a higher priority and any values set through the <device/> parameter in a configuration file will have a lower priority.

Domain Name System (DNS) Parameters

Parameter	*Permitted Values*	*Default*
tcpIpApp.dns.server[1]	**Dotted-decimal IP address**	**Null**
The primary server to which the phone directs DNS queries.		
tcpIpApp.dns.altServer[1]	**Dotted-decimal IP address**	**Null**
The secondary server to which the phone directs DNS queries.		
tcpIpApp.dns.domain[1]	**String**	**Null**
The phone's DNS domain.		

[1] Change causes phone to restart or reboot.

<ice/>

The <ice/> parameters in the table Ice Parameters enable you to set the STUN/TURN/ICE feature.

Ice Parameters

Parameter	*Permitted Values*	*Default*
tcpIpApp.ice.mode	**Disabled, Standard, MSOCS**	**Disabled**
Turn SIP ICE negotiation on or off. If using Lync Server 2010, set to MSOCS to enable ICE.		
tcpIpApp.ice.password	**String**	**Null**
Enter the password to authenticate to the TURN server.		
tcpIpApp.ice.stun.server	**String**	**Null**
Enter the IP address of the STUN server.		
tcpIpApp.ice.stun.udpPort	**1-65535**	**3478**
The UDP port number of the STUN server.		
tcpIpApp.ice.tcp.enabled	**0 or 1**	**1**
If 0, TCP is disabled. If 1, TCP is enabled.		
tcpIpApp.ice.turn.callAdmissionControl.enabled		**1**
tcpIpApp.ice.turn.server	**String**	**Null**
Enter the IP address of the TURN server.		
tcpIpApp.ice.turn.tcpPort	**1-65535**	**443**
The UDP port number of the TURN server.		
tcpIpApp.ice.turn.udpPort	**1-65535**	**443**
The UDP port number of the TURN server.		
tcpIpApp.ice.username	**String**	**Null**
Enter the user name to authenticate to the TURN server.		

<sntp/>

The table Simple Network Time Protocol (SNTP) Parameters lists the Simple Network Time Protocol (SNTP) parameters used to set up time synchronization and daylight savings time. The default values will enable and configure daylights savings time (DST) for North America.

Daylight savings time defaults:

- Do not use fixed day, use first or last day of week in the month.

- Start DST on the second Sunday in March at 2am.
- Stop DST on the first Sunday in November at 2am.

Simple Network Time Protocol (SNTP) Parameters

Parameter	*Permitted Values*	*Default*
tcpIpApp.sntp.address	**Valid hostname or IP address**	**Null**
The address of the SNTP server.		
tcpIpApp.sntp.address.overrideDHCP	**0 or 1**	**0**
If 0, the DHCP values for the SNTP server address will be used. If 1, the SNTP parameters will override the DHCP values.		
tcpIpApp.sntp.daylightSavings.enable	**0 or 1**	**1**
If 0, daylight savings time rules are not applied to the displayed time. If 1, the daylight savings rules apply.		
tcpIpApp.sntp.daylightSavings.fixedDayEnable	**0 or 1**	**0**
If 0, `month`, `date`, and `dayOfWeek` are used in the DST calculation. If 1, only `month` and `date` are used.		
tcpIpApp.sntp.daylightSavings.start.date	**1 to 31**	**8**
The start date for daylight savings time. If `fixedDayEnable` is set to 1, the value of this parameter is the day of the month to start DST. If `fixedDayEnable` is set to 0, this value specifies the occurrence of `dayOfWeek` when DST should start. Set 1 for the first occurrence in the month, set 8 for the second occurrence, 15 for the third occurrence, or 22 for the fourth occurrence. For example, if set to 15, DST starts on the third `dayOfWeek` of the month.		
tcpIpApp.sntp.daylightSavings.start.dayOfWeek	**1 to 7**	**1**
The day of the week to start DST. 1=Sunday, 2=Monday, … 7=Saturday. Note: this parameter is not used if `fixedDayEnable` is set to 1.		
tcpIpApp.sntp.daylightSavings.start.dayOfWeek.lastInMonth	**0 or 1**	**0**
If 1, DST starts on the last `dayOfWeek` of the month and the `start.date` is ignored). Note: this parameter is not used if `fixedDayEnable` is set to 1.		
tcpIpApp.sntp.daylightSavings.start.month	**1 to 12**	**3 (March)**
The month to start DST. 1=January, 2=February… 12=December.		
tcpIpApp.sntp.daylightSavings.start.time	**0 to 23**	**2**
The time of day to start DST – in 24 hour clock format. 0= 12am, 1= 1am,… 12= 12pm, 13= 1pm, … 23= 11pm.		
tcpIpApp.sntp.daylightSavings.stop.date	**1 to 31**	**1**
The stop date for daylight savings time. If `fixedDayEnable` is set to 1, the value of this parameter is the day of the month to stop DST. If `fixedDayEnable` is set to 0, this value specifies the occurrence of `dayOfWeek` when DST should stop. Set 1 for the first occurrence in the month, set 8 for the second occurrence, 15 for the third occurrence, or 22 for the fourth occurrence. For example, if set to 22, DST stops on the fourth `dayOfWeek` of the month.		

Parameter	*Permitted Values*	*Default*
tcpIpApp.sntp.daylightSavings.stop.dayOfWeek	**1 to 7**	**1**
The day of the week to stop DST. 1=Sunday, 2=Monday, … 7=Saturday. Note: this parameter is not used if `fixedDayEnable` is set to 1.		
tcpIpApp.sntp.daylightSavings.stop.dayOfWeek.lastInMonth	**0 or 1**	**0**
If 1, DST stops on the last `dayOfWeek` of the month and the `stop.date` is ignored). Note: this parameter is not used if `fixedDayEnable` is set to 1.		
tcpIpApp.sntp.daylightSavings.stop.month	**1 to 12**	**11**
The month to stop DST. 1=January, 2=February… 12=December.		
tcpIpApp.sntp.daylightSavings.stop.time	**0 to 23**	**2**
The time of day to stop DST – in 24 hour clock format. 0= 12am, 1= 1am,… 12= 12pm, 13= 1pm, … 23= 11pm.		
tcpIpApp.sntp.gmtOffset	**positive or negative integer**	**0**
The offset in seconds of the local time zone from GMT.3600 seconds = 1 hour, -3600 seconds = -1 hour.		
tcpIpApp.sntp.gmtOffset.overrideDHCP	**0 or 1**	**0**
If 0, the DHCP values for the GMT offset will be used. If 1, the SNTP values for the GMT offset will be used.		
tcpIpApp.sntp.resyncPeriod	**positive integer**	**86400**
The period of time (in seconds) that passes before the phone resynchronizes with the SNTP server. Note: 86400 seconds is 24 hours.		

<port/><rtp/>

The parameters listed in the table RTP Port Parameters enable you to configure the port filtering used for RTP traffic.

RTP Port Parameters

Parameter	*Permitted Values*	*Default*
tcpIpApp.port.rtp.filterByPort[1]	**0 or 1**	**0**
Ports can be negotiated through the SDP protocol. If set to 1, the phone will reject RTP packets arriving from (sent from) a non-negotiated port.		
tcpIpApp.port.rtp.forceSend[1]	**0 to 65535**	**0**
Send all RTP packets to, and expect all RTP packets to arrive on, this port. If 0, RTP traffic is not forced to one port. Note: Both `tcpIpApp.port.rtp.filterByIp` and `tcpIpApp.port.rtp.filterByPort` must be set to 1 for this to work.		

Parameter	Permitted Values	Default
tcpIpApp.port.rtp.mediaPortRangeStart[1]	**even integer 1024 to 65440**	**2222**
The starting port for RTP packets. Ports will be allocated from a pool starting with this port up to a value of (start-port + 47) for a voice-only phone or (start-port + 95) for a video phone. Note: Ensure that there is no contention for port numbers. For example, do not use 5060 (default port for SIP).		

[1] Change causes phone to restart or reboot.

<keepalive/>

The parameters listed in the table TCP Keep-Alive Parameters enable the configuration of TCP keep-alive on SIP TLS connections; the phone can detect a failure quickly (in minutes) and attempt to re-register with the SIP call server (or its redundant pair).

TCP Keep-Alive Parameters

Parameter	Permitted Values	Default
tcpIpApp.keepalive.tcp.idleTransmitInterval	**10 to 7200**	**30**
The amount of time to wait (in seconds) before sending the keep-alive message to the call server. Note: If this parameter is set to a value that is out of range, the default value is used.		
tcpIpApp.keepalive.tcp.noResponseTransmitInterval	**5 to 120**	**20**
If no response is received to a keep-alive message, subsequent keep-alive messages are sent to the call server at this interval (every x seconds).		
tcpIpApp.keepalive.tcp.sip.tls.enable	**0 or 1**	**0**
If 0, disable TCP keep-alive for SIP signaling connections that use TLS transport. If 1, enable TCP keep-alive for SIP signaling connections that use TLS transport.		

[1] Change causes phone to restart or reboot.

<fileTransfer/>

The parameters listed in the table File Transfer Parameters provide information on file transfers from the phone to the Provisioning server.

File Transfer Parameters

Parameter	Permitted Values	Default
tcpIpApp.fileTransfer.waitForLinkIfDown	**0 or 1**	**1**
If 1, file transfer from the FTP server is delayed until Ethernet comes back up. If 0, file transfer from the FTP server is not attempted.		

<tones/>

This parameter describes configuration items for the tone resources available in the phone. It includes:

- <DTMF/>
- <chord/>

<DTMF/>

The parameters listed in the table DTMF Tone Parameters enable you to configure Dual-tone multi-frequency (DTMF) tone signaling.

DTMF Tone Parameters

Parameter	*Permitted Values*	*Default*
tone.dtmf.chassis.masking[1]	**0 or 1**	**0**
If 0, DTMF tones will be played through the speakerphone in handsfree mode. If 1 (set only if `tone.dtmf.viaRtp` is set to 0), DTMF tones will be substituted with non-DTMF pacifier tones when dialing in handsfree mode—this is to prevent the tones from broadcasting to surrounding telephony devices or being inadvertently transmitted in-band due to local acoustic echo.		
tone.dtmf.level[1]	**-33 to 3**	**-15**
The level of the high frequency component of the DTMF digit measured in dBm0; the low frequency tone will be two dB lower.		
tone.dtmf.offTime[1]	**positive integer**	**50**
When a sequence of DTMF tones is played out automatically, this is the length of time in milliseconds the phone will pause between digits. This is also the minimum inter-digit time when dialing manually.		
tone.dtmf.onTime[1]	**positive integer**	**50**
When a sequence of DTMF tones is played out automatically, this is the length of time in milliseconds the tones will be played for. This is also the minimum time the tone will be played when dialing manually (even if key press is shorter).		
tone.dtmf.rfc2833Control[1]	**0 or 1**	**1**
If set to 1, the phone will indicate a preference for encoding DTMF through RFC 2833 format in its Session Description Protocol (SDP) offers by showing support for the phone-event payload type. This does not affect SDP answers; these will always honor the DTMF format present in the offer since the phone has native support for RFC 2833.		
tone.dtmf.rfc2833Payload[1]	**96 to 127**	**127**
The phone-event payload encoding in the dynamic range to be used in SDP offers.		
tone.dtmf.viaRtp[1]	**0 or 1**	**1**
If set to 1, encode DTMF in the active RTP stream. Otherwise, DTMF may be encoded within the signaling protocol only when the protocol offers the option. Note: If this parameter is set to 0, `tone.dtmf.chassis.masking` should be set to 1.		

Parameter	*Permitted Values*	*Default*

[1] Change causes phone to restart or reboot.

<chord/>

Chord-sets are the building blocks of sound effects that used synthesized audio rather than sampled audio. Most call progress and ringer sound effects are synthesized. A chord-set is a multi-frequency note with an optional on/off cadence. A chord-set can contain up to four frequency components generated simultaneously, each with its own level. Chord parameters are listed in the table Chord Parameters.

There are three chord sets: callProg, misc, and ringer. Each chord set has different chord names, represented by *x* in the following table. The chord names are as follows:

For **callProg**, *x* can be one of the following chords:

- **dialTone**, **busyTone**, **ringback**, **reorder**, **stutter_3**, **callWaiting**, **callWaitingLong**, **howler**, **recWarning**, **stutterLong**, **intercom**, **callWaitingLong**, **precedenceCallWaiting**, **preemption**, **precedenceRingback**, or **spare1** to **spare6**.

For **misc**, *x* can be one of the following chords

- **spare1** to **spare9**.

For **ringer,** *x* can be one of the following chords:

- **ringback**, **originalLow**, **originalHigh**, or **spare1** to **spare19**.

Chord Parameters

Parameter	*Permitted Values*
tone.chord.callProg.x.freq.y	**0-1600**
tone.chord.misc.x.freq.y	**0-1600**
tone.chord.ringer.x.freq.y	**0-1600**
The frequency (in Hertz) for component y. Up to six chord-set components can be specified (y=1 to 6).	
tone.chord.callProg.x.level.y	**-57 to 3**
tone.chord.misc.x.level.y	**-57 to 3**
tone.chord.ringer.x.level.y	**-57 to 3**
The level of component y in dBm0. Up to six chord-set components can be specified (y=1 to 6).	
tone.chord.callProg.x.onDur	**positive integer**
tone.chord.misc.x.onDur	**positive integer**
tone.chord.ringer.x.onDur	**positive integer**
The on duration (length of time to play each component) in milliseconds, 0=infinite.	
tone.chord.callProg.x.offDur	**positive integer**
tone.chord.misc.x.offDur	**positive integer**
tone.chord.ringer.x.offDur	**positive integer**
The off duration (the length of silence between each chord component) in milliseconds, 0=infinite.	

Parameter	*Permitted Values*
tone.chord.callProg.x.repeat	**positive integer**
tone.chord.misc.x.repeat	**positive integer**
tone.chord.ringer.x.repeat	**positive integer**
The number of times each ON/OFF cadence is repeated, 0=infinite.	

<up/>

Use the parameters listed in the table User Preferences Parameters to set user preferences on the phones.

User Preferences Parameters

Parameter	*Permitted Values*	*Default*
up.25mm	**1 or 2**	**1**
Specify whether to use a mobile phone or a PC to connect to the 2.5mm audio port on a conference phone. Set to 1 if using a mobile phone. Set to 2 if using a PC.		
up.analogHeadsetOption	**0, 1, 2, 3**	**0**
The Electronic Hookswitch mode for the phone's analog headset jack. 0 - no EHS-compatible headset is attached. 1 - a Jabra EHS-compatible headset is attached. 2 - a Plantronics EHS-compatible headset is attached. 3 - a Sennheiser EHS-compatible headset is attached.		
up.audioMode	**0 or 1**	**0**
If 0, a handset is connected. If 1, a headset is connected.		
up.backlight.idleIntensity	**0, 1, 2, or 3**	**1**
The brightness of the LCD backlight when the phone is idle. 0 – off, 1 – low, 2 – medium, 3 – high. Note: If this is higher than the active backlight brightness (`onIntensity`), the active backlight brightness is used.		
up.backlight.onIntensity	**0, 1, 2, or 3**	**3**
The brightness of the LCD backlight when the phone is active (in use). 0: off, 1 – low, 2 – medium, 3 – high		
up.backlight.timeout	**5 to 60**	**40**
The number of seconds to wait before the backlight dims from the active intensity to the idle intensity.		
up.cfgWarningsEnabled	**0 or 1**	**0**
If 1, a warning is displayed on the phone if the phone is configured with pre-UC software 3.3.0 parameters. If 0, the warning will not display.		
up.handsfreeMode	**0 or 1**	**1**
If 0, the handsfree speakerphone is disabled (cannot be used). If 1, the handsfree speakerphone is enabled.		

Parameter	*Permitted Values*	*Default*
up.headsetAlwaysUseIntrinsicRinger	**0 or 1**	**1**
If 1, the USB headset will use the intrinsic ringer mixed with DSP ringer when the sound effect destination is the USB headset.		
up.headsetMode	**0 or 1**	**0**
If 0, handsfree mode will be used by default instead of the handset. If 1, the headset will be used as the preferred audio mode after the headset key is pressed for the first time, until the headset key is pressed again.		
up.headset.phoneVolumeControl[1]	**disable, enable, auto**	**auto**
Controls the phone’s behavior when you adjust volume at the headset. **enable** – The phone responds to volume up/down events from the headset by displaying the volume widget in the phone’s user interface and adjusting the phone's internal volume. **disable** – The phone ignores volume up/down events from the headset; pressing the headset's volume controls has no effect on the phone. **auto** – The phone automatically selects which of the above two behaviors to apply based on the type and model of headset that you attach.		
up.hearingAidCompatibility.enabled	**0 or 1**	**0**
If set to 1, the phone audio Rx (receive) equalization is disabled for hearing aid compatibility. If 0, audio Rx equalization is enabled.		
up.idleBrowser.enabled	**0 or 1**	**0**
If 0, the idle browser is disabled. If 1, the idle browser is enabled (if `up.prioritizeBackground.enable` is 1, the user can choose to display the background or the idle browser through the phone menu).		
up.idleStateView[1]	**0 or 1**	**0**
Sets the default view on the phone. If 0, The call/line view is the default view. If 1, the Home screen is the default view.		
up.idleTimeout[1]	**0 to 65535, seconds**	**40**
The number of seconds that the phone can be idle for before automatically leaving a menu and showing the idle display. If 0, there is no timeout and the phone does not automatically exit to the idle display.		
up.IdleViewPreferenceRemoteCalls1	**0 or 1**	**0**
Use this parameter to determine when the phone displays the idle browser. When set to 1, a phone with only remote calls active, is treated as in the active state and the idle browser does not display. When set to 0, a phone with only remote calls active, is treated as in the idle state and the idle browser displays.		
up.lineKeyCallTerminate	**0 or 1**	**0**
If 1, the user can press a line key to end an active call on that line. If 0, the user cannot end a call by pressing the line key (this is the previous behavior).		
up.localClockEnabled	**0 or 1**	**1**
If 0, the date and time are not shown on the idle display. If 1, the date and time and shown on the idle display.		

Parameter	*Permitted Values*	*Default*
up.mwiVisible[1]	**0 or 1**	**0**
If set is 0, the incoming MWI notifications for lines where the MWI callback mode is disabled (`msg.mwi.x.callBackMode` is set to 0) are ignored, and do not appear in the message retrieval menus. If set to 1, the MWI for lines whose MWI is disabled will display (pre-SIP 2.1 behavior), even though MWI notifications have been received for those lines.		
up.numberFirstCID[1]	**0 or 1**	**0**
If 0, the caller ID display will show the caller's name first. If 1, the caller's phone number will be shown first.		
up.numOfDisplayColumns	**1, 2, 3, 4**	**max 4**
Set the maximum number of columns the CX5500 system displays. Note that phones display one column when the value is set to 0. The maximum number of columns for the CX5500 system is 4.		
up.offHookAction.none[1]	**0 or 1**	**0**
If 0, the behavior will be as it was in SIP 2.1.2. If 1, when the user lifts the handset, the phone will not seize the line and the ringer will continue until the user takes further action.		
up.oneTouchVoiceMail[1]	**0 or 1**	**0**
If set to 1, the voicemail summary display is bypassed and voicemail is dialed directly (if configured).		
up.screenCapture.enabled[1]	**0 or 1**	**0**
If 0, screen captures are disabled. If 1, the user can enable screen captures from the Screen Capture menu on the phone. Note: when the phone reboots, screen captures are disabled from the Screen Capture menu on the phone.		
up.screenSaver.enabled	**0 or 1**	**0**
. If 0, the screen saver feature is disabled. If 1, the screen saver feature is enabled. If a USB flash drive containing images is connected to the phone, and the idle browser is not configured, a slide show will cycle through the images from the USB flash drive when the screen saver feature is enabled. The images must be stored in the directory on the flash drive specified by `up.pictureFrame.folder`. The screen saver displays when the phone has been in the idle state for the amount of time specified by `up.screenSaver.waitTime`.		
up.screenSaver.type	**0, 1, 2**	**0**
The type of screen saver. If 0, the screen saver feature is disabled. If 1, a blank screen is used, If 2, the idle browser is used as the screen saver.		
up.screenSaver.waitTime	**1 to 9999, minutes**	**15**
. The number of minutes that the phone waits in the idle state before the screen saver starts.		
up.simplifiedSipCallInfo	**0 or 1**	**0**
If 1, the displayed host name is trimmed for both incoming and outgoing calls and the protocol tag/information is not displayed for incoming and outgoing calls.		
up.useDirectoryNames[1]	**0 or 1**	**1**
If 0, names provided through network signaling are used for caller ID. If 1, the name field in the local contact directory will be used as the caller ID for incoming calls from contacts in the local directory. Note: Outgoing calls and corporate directory entries are not matched.		

Parameter	*Permitted Values*	*Default*
up.warningLevel[1]	**0 to 2**	**0**
If 0, the phone's warning icon and a pop-up message display on the phone for all warnings. If 1, the warning icon and pop-up messages are only shown for critical warnings. Note: All warnings are listed in the Warnings menu (navigate to **Settings > Status > Diagnostics > Warnings** on the phone).		
up.welcomeSoundEnabled[1]	**0 or 1**	**1**
If 0, the welcome sound is disabled. If 1, the welcome sound is enabled and played each time the phone reboots.		
up.welcomeSoundOnWarmBootEnabled[1]	**0 or 1**	**0**
If 0, the welcome sound is played when the phone powers up (cold boot), but not after it restarts or reboots (warm boot). If 1, the welcome sound plays each time the phone powers up, reboots, or restarts.		

[1] Change causes phone to restart or reboot.

<upgrade/>

Use the parameters listed in the table Upgrade Server Parameters to specify the URL of a custom download server and the Polycom UC Software download server for the phone to check when searching for software upgrades.

Upgrade Server Parameters

Parameter	*Permitted Values*	*Default*
upgrade.custom.server.url	**URL**	**Null**
The URL of a custom download server.		
upgrade.plcm.server.url	**URL**	**http://downloads.polycom.com/voice/software/**
The URL of the Polycom UC Software download server.		

<video/>

The parameters in the table General Video Parameters are supported on the CX5500 system.

This parameter also includes:

- <camera/>
- <codecs/>

General Video Parameters

Parameter	*Permitted Values*	*Default*
video.autoFullScreen	**0 or 1**	**0**
If 0, video calls only use the full screen layout if it is explicitly selected by the user. If 1, video calls use the full screen layout by default, such as when a video call is first created or when an audio call transitions to a video call)		
video.autoStartVideoTx	**0 or 1**	**1**
When enabled, video transmission to the far side begins when you start a call. When disabled, video transmission does not begin until you press the **Video > Start Video** soft keys. This parameter controls video sent to the far side. Video from the far side will always be displayed if it is available, and far side users can control when to send video.		
video.callMode.default	**audio or video**	**audio**
Allows the user to select the mode to use when using SIP protocol only.		
video.callRate	**128 to 2048**	**512**
The default call rate (in kbps) to use when initially negotiating bandwidth for a video call.		
video.dynamicControlMethod	**0 or 1**	**1**
If 1, the first I-Frame request uses the method defined by `video.forceRtcpVideoCodecControl` and subsequent requests alternate between RTCP-FB and SIP INFO.		
video.enable	**0=Disable, 1=Enable**	**1**
If 0, video is not enabled and all calls—both sent and received—are audio-only. If 1, video is sent in outgoing calls and received in incoming calls if the other device supports video.		
video.iFrame.delay[1]	**0 to 10, seconds**	**0**
When non-zero, an extra I-frame is transmitted after video starts. The amount of delay from the start of video until the I-frame is sent is configurable up to 10 seconds. Use a value of 2 seconds if you are using this parameter in a Microsoft Lync environment.		
video.iFrame.minPeriod	**1 - 60**	**2**
After sending an I-frame, the phone will always wait at least this amount of time before sending another I-frame in response to requests from the far end.		
video.iFrame.onPacketLoss	**0 or 1**	**0**
If 1, an I-frame is transmitted to the far end when a received RTCP report indicates that video RTP packet loss has occurred.		
video.maxCallRate[1]	**128 to 2048 kbps**	**768**
The maximum call rate allowed. This allows the administrator to limit the maximum call rate that the users can select. If `video.callRate` exceeds this value, this value will be used as the maximum.		
video.quality[1]	**motion, sharpness**	**Null**
The optimal quality for video that you send in a call or a conference. Use `motion` if your outgoing video will have motion or movement. Use `sharpness` or Null if your outgoing video will have little or no movement. Note: If `motion` is not selected, moderate to heavy motion can cause some frames to be dropped.		

Parameter	*Permitted Values*	*Default*
video.screenMode	**normal, full, crop**	**normal**
The screen mode for the video window shown in non-full screen mode. If set to `normal` or Null, the entire view is displayed and horizontal or vertical black bars may appear on the edges to maintain the correct aspect ratio. If set to `full`, the entire view is stretched linearly and independently to fill the video frame. If set to `crop`, black bars are not shown, the image is re-sized and enlarged to cover the entire video frame, and parts of the image that do not fit in the display are cropped (removed).		
video.screenModeFS	**normal, full, crop**	**normal**
The screen mode for the video window shown in full screen mode. If set to `normal` or Null, the entire view is displayed and horizontal or vertical black bars may appear on the edges to maintain the correct aspect ratio. If set to `full`, the entire view is stretched linearly and independently to fill the screen. If set to `crop`, black bars are not shown, the image is re-sized and enlarged to cover the entire screen, and parts of the image that do not fit in the display are cropped (removed).		

[1] Change causes phone to restart or reboot.

<camera/>

The settings in the table Video Camera Parameters control the performance of the camera.

Video Camera Parameters

Parameter	*Permitted Values*	*Default*
video.camera.brightness	**0 to 6**	**3**
Set brightness level. The value range is from 0 (Dimmest) to 6 (Brightest).		
video.camera.contrast	**0 to 4**	**0**
Set contrast level. The value range is from 0 (No contrast increase) to 3 (Most contrast increase), and 4 (Noise reduction contrast).		
video.camera.flickerAvoidance	**0 to 2**	**0**
Set flicker avoidance. If set to 0, flicker avoidance is automatic. If set to 1, 50hz AC power frequency flicker avoidance (Europe/Asia). If set to 2, 60hz AC power frequency flicker avoidance (North America).		
video.camera.frameRate	**5 to 30**	**25**
Set target frame rate (frames per second). Values indicate a fixed frame rate, from 5 (least smooth) to 30 (most smooth). Note: If `video.camera.frameRate` is set to a decimal number, the value 25 is used.		
video.camera.saturation	**0 to 6**	**3**
Set saturation level. The value range is from 0 (Lowest) to 6 (Highest).		

Parameter	*Permitted Values*	*Default*
video.camera.sharpness	**0 to 6**	**3**
Set sharpness level. The value range is from 0 (Lowest) to 6 (Highest).		

<codecs/>

These video codecs include:

- <profile/>

<profile/>

The table Video Profile Parameters contains settings for a group of low-level video codec parameters. For most use cases, the default values will be appropriate. Polycom does not recommend changing the default values unless specifically advised to do so.

Video Profile Parameters

Parameter	*Permitted Values*
video.profile.H261.annexD[1]	**0 or 1 (default)**
Enable or disable Annex D when negotiating video calls.	
video.profile.H261.CifMpi[1]	**1 (default) to 32**
Specify the frame rate divider that the phone uses when negotiating CIF resolution for a video call. You can enter a value between 0-4. To disable, enter '0'. The default frame rate divider is '1'.	
video.profile.H261.jitterBufferMax[1]	**(video.profile.H261.jitter BufferMin + 500ms) to 2500ms, default 2000ms**
The largest jitter buffer depth to be supported (in milliseconds). Jitter above this size will always cause lost packets. This parameter should be set to the smallest possible value that will support the expected network jitter.	
video.profile.H261.jitterBufferMin[1]	**33ms to 1000ms, default 150ms**
The smallest jitter buffer depth (in milliseconds) that must be achieved before play out begins for the first time. Once this depth has been achieved initially, the depth may fall below this point and play out will still continue. This parameter should be set to the smallest possible value which is at least two packet payloads, and larger than the expected short term average jitter.	
video.profile.H261.jitterBufferShrink[1]	**33ms to 1000ms, default 70ms**
The absolute minimum duration time (in milliseconds) of RTP packet Rx with no packet loss between jitter buffer size shrinks. Use smaller values (33 ms) to minimize the delay on known good networks. Use larger values (1000ms) to minimize packet loss on networks with large jitter (3000 ms).	
video.profile.H261.payloadType[1]	**0 to 127, default 31**
RTP payload format type for H261 MIME type.	

Parameter	*Permitted Values*
video.profile.H261.QcifMpi[1]	**1 (default) to 32**
Specify the frame rate divider that the phone uses when negotiating Quarter CIF resolution for a video call. You can enter a value between 0-4. To disable, enter '0'. The default frame rate divider is '1'.	
video.profile.H263.CifMpi[1]	**1 (default) to 32**
Specify the frame rate divider that the phone uses when negotiating CIF resolution for a video call. You can enter a value between 0-32. To disable, enter '0'. The default frame rate divider is '1'.	
video.profile.H263.jitterBufferMax[1]	**(video.profile.H263.jitter BufferMin + 500ms) to 2500ms, default 2000ms**
The largest jitter buffer depth to be supported (in milliseconds). Jitter above this size will always cause lost packets. This parameter should be set to the smallest possible value that will support the expected network jitter.	
video.profile.H263.jitterBufferMin[1]	**33ms to 1000ms, default 150ms**
The smallest jitter buffer depth (in milliseconds) that must be achieved before play out begins for the first time. Once this depth has been achieved initially, the depth may fall below this point and play out will still continue. This parameter should be set to the smallest possible value which is at least two packet payloads, and larger than the expected short term average jitter.	
video.profile.H263.jitterBufferShrink[1]	**33ms to 1000ms, default 70ms**
The absolute minimum duration time (in milliseconds) of RTP packet Rx with no packet loss between jitter buffer size shrinks. Use smaller values (33 ms) to minimize the delay on known good networks. Use larger values (1000ms) to minimize packet loss on networks with large jitter (3000 ms).	
video.profile.H263.payloadType[1]	**0 to 127, default 34**
RTP payload format type for H263 MIME type.	
video.profile.H263.QcifMpi[1]	**1 (default) to 32**
Specify the frame rate divider that the phone uses when negotiating Quarter CIF resolution for a video call. You can enter a value between 0-32. To disable, enter '0'. The default frame rate divider is '1'.	
video.profile.H263.SqcifMpi[1]	**1 (default) to 32**
Specify the frame rate divider that the phone uses when negotiating Sub Quarter CIF resolution for a video call. You can enter a value between 0-32. To disable, enter '0'. The default frame rate divider is '1'.	
video.profile.H2631998.annexF[1]	**0 (default) or 1**
Enable or disable Annex F when negotiating video calls.	
video.profile.H2631998.annexI[1]	**0 (default) or 1**
Enable or disable Annex I when negotiating video calls.	
video.profile.H2631998.annexJ[1]	**0 (default) or 1**
Enable or disable Annex J when negotiating video calls.	
video.profile.H2631998.annexK[1]	**0, 1 (default), 2, 3, 4**
Specify the value of Annex K to use when negotiating video calls. You can enter a value between 0-4. To disable, enter '0'. The default value is '1'.	

Parameter	*Permitted Values*
video.profile.H2631998.annexN[1]	**0, 1 (default), 2, 3, 4**
Specify the value of Annex N to use when negotiating video calls. You can enter a value between 0-4. To disable, enter '0'. The default value is '1'.	
video.profile.H2631998.annexT[1]	**0 (default) or 1**
Enable or disable Annex T when negotiating video calls.	
video.profile.H2631998.CifMpi[1]	**1 (default) to 32**
Specify the frame rate divider that the phone uses when negotiating CIF resolution for a video call. You can enter a value between 0-32. To disable, enter '0'. The default frame rate divider is '1'.	
video.profile.H2631998.jitterBufferMax[1]	**(video.profile.H2631998.ji tterBufferMin+ 500ms) to 2500ms, default 2000ms**
The largest jitter buffer depth to be supported (in milliseconds). Jitter above this size will always cause lost packets. This parameter should be set to the smallest possible value that will support the expected network jitter.	
video.profile.H2631998.jitterBufferMin[1]	**33ms to 1000ms, default 150ms**
The smallest jitter buffer depth (in milliseconds) that must be achieved before play out begins for the first time. Once this depth has been achieved initially, the depth may fall below this point and play out will still continue. This parameter should be set to the smallest possible value which is at least two packet payloads, and larger than the expected short term average jitter.	
video.profile.H2631998.jitterBufferShrink[1]	**33ms to 1000ms, default 70ms**
The absolute minimum duration time (in milliseconds) of RTP packet Rx with no packet loss between jitter buffer size shrinks. Use smaller values (33 ms) to minimize the delay on known good networks. Use larger values (1000ms) to minimize packet loss on networks with large jitter (3000 ms).	
video.profile.H2631998.payloadType[1]	**96 (default) to 127**
RTP payload format type for H263-1998/90000 MIME type.	
video.profile.H2631998.QcifMpi[1]	**1 (default) to 32**
Specify the frame rate divider that the phone uses when negotiating Quarter CIF resolution for a video call. You can enter a value between 0-32. To disable, enter '0'. The default frame rate divider is '1'.	
video.profile.H2631998.SqcifMpi[1]	**1 (default) to 32**
Specify the frame rate divider that the phone uses when negotiating Sub Quarter CIF resolution for a video call. You can enter a value between 0-32. To disable, enter '0'. The default frame rate divider is '1'.	
video.profile.H264.jitterBufferMax[1]	**(video.profile.H264.jitter BufferMin + 500ms) to 2500ms, default 2000ms**
The largest jitter buffer depth to be supported (in milliseconds). Jitter above this size will always cause lost packets. This parameter should be set to the smallest possible value that will support the expected network jitter.	
video.profile.H264.jitterBufferMin[1]	**33ms to 1000ms, default 150ms**
The smallest jitter buffer depth (in milliseconds) that must be achieved before play out begins for the first time. Once this depth has been achieved initially, the depth may fall below this point and play out will still continue. This parameter should be set to the smallest possible value which is at least two packet payloads, and larger than the expected short term average jitter.	

Parameter	*Permitted Values*
video.profile.H264.jitterBufferShrink[1]	**33ms to 1000ms, default 70ms**
The absolute minimum duration time (in milliseconds) of RTP packet Rx with no packet loss between jitter buffer size shrinks. Use smaller values (33 ms) to minimize the delay on known good networks. Use larger values (1000ms) to minimize packet loss on networks with large jitter (3000 ms).	
video.profile.H264.payloadType[1]	**96 to 127, default 109**
RTP payload format type for H264/90000 MIME type.	
video.profile.H264.profileLevel[1]	**1, 1b, 1.1, 1.2, and 1.3 (default)**
Specify the highest profile level within the Baseline profile supported in video calls. The phone supports the following levels: 1, 1b, 1.1, 1.2, 1.3. The default level is 1.3. For more information, refer to ITU-T H.264.	

[1] Change causes phone to restart or reboot.

<voice/>

The parameters listed in the table Voice Parameters control audio on the phone.

Voice Parameters

Parameter	*Permitted Values*	*Default*
voice.txPacketDelay[1]	**low, normal, Null**	**Null**
If set to normal or Null, no audio parameters are changed. If set to low and there are no precedence conflicts, the following changes are made: • `voice.codecPref.G722="1"` • `voice.codecPref.G711Mu="2"` • `voice.codecPref.G711A="3"` • `voice.codecPref.<OtherCodecs>=""` • `voice.audioProfile.G722.payloadSize="10"` • `voice.audioProfile.G711Mu.payloadSize= "10"` • `voice.audioProfile.G711A.payloadSize= "10"` • `voice.aec.hs.enable="0"` • `voice.ns.hs.enable="0"`		
voice.txPacketFilter[1]	**0 or 1**	**Null**
If 0, no Tx filtering is performed. If 1, narrowband Tx high pass filter is enabled.		

[1] Change causes phone to restart or reboot.

This parameter includes:

- <codecPref/>
- <volume/>
- <vad/>

- <quality monitoring/>
- <rxQoS/>

<codecPref/>

As of Polycom UC Software 3.3.0, you can configure a simplified set of codec properties for all phone models to improve consistency and reduce workload on the phones. Phone codec preferences are listed in the table Voice Codec Preferences Parameters.

If a particular phone does not support a codec, the phone will ignore that codec and continue to the codec next in the priority.

For more information on codecs on particular phones and priorities, see Audio Codecs.

Voice Codec Preferences Parameters

Parameter	*Permitted Values*	*Default*
voice.codecPref.G711_A	**0 to 27**	**7**
voice.codecPref.G711_Mu		**6**
voice.codecPref.G719.32kbps		**0**
voice.codecPref.G719.48kbps		**0**
voice.codecPref.G719.64kbps		**0**
voice.codecPref.G722		**4**
voice.codecPref.G7221.16kbps		**0**
voice.codecPref.G7221.24kbps		**0**
voice.codecPref.G7221.32kbps		**5**
voice.codecPref.G7221_C.24kbps		**0**
voice.codecPref.G7221_C.32kbps		**0**
voice.codecPref.G7221_C.48kbps		**2**
voice.codecPref.G729_AB		**8**
voice.codecPref.iLBC.13_33kbps		**0**
voice.codecPref.iLBC.15_2kbps		**0**
voice.codecPref.Lin16.8ksps		**0**
voice.codecPref.Lin16.16ksps		**0**
voice.codecPref.Lin16.32ksps		**0**
voice.codecPref.Lin16.44_1ksps		**0**
voice.codecPref.Lin16.48ksps		**0**
voice.codecPref.Siren14.24kbps		**0**
voice.codecPref.Siren14.32kbps		**0**
voice.codecPref.Siren14.48kbps		**3**
voice.codecPref.Siren22.32kbps		**0**
voice.codecPref.Siren22.48kbps		**0**
voice.codecPref.Siren22.64kbps		**0**
The priority of the codec. If 0 or Null, the codec is disabled. A value of 1 is the highest priority. If a phone does not support a codec, it will treat the setting as if it were 0 and not offer or accept calls with that codec.		

<volume/>

In some countries, regulations state that a phone’s receiver volume must be reset to a nominal level for each new call. This is the phone’s default behavior. Using the parameters listed in the table Volume Parameters, you can set the receiver volume to persist across calls each time a user makes changes to the default volume level.

Volume Parameters

Parameter	*Permitted Values*	*Default*
voice.volume.persist.handset[1]	**0 or 1**	**0**
If 0, the handset receive volume will automatically reset to a nominal level after each call. If 1, the volume for each call will be the same as the previous call. If set to 1, the handset receive volume will persist across calls. If set to 0, the handset receive volume will be reset to nominal at the start of each call.		
voice.volume.persist.headset[1]	**0 or 1**	**0**
If 0, the headset receive volume will automatically rest to a nominal level after each call. If 1, the volume for each call will be the same as the previous call.		
voice.volume.persist.handsfree[1]	**0 or 1**	**1**
If 0, the speakerphone receive volume will automatically rest to a nominal level after each call. If 1, the volume for each call will be the same as the previous call.		
voice.volume.persist.usb.handsfree[1]	**0 or 1**	**1**
If 0, the USB headset will not be used. If 1, the USB headset will be used in handsfree mode.		
voice.volume.persist.usbHeadset[1]	**0 or 1**	**0**
If 0, the USB headset will not be used. If 1, the USB headset will be used.		

[1] Change causes phone to restart or reboot.

<vad/>

The paramters listed in Voice Activity Detection Parameters control the performance of the voice activity detection (silence suppression) feature.

Voice Activity Detection (VAD) Parameters

Parameter	*Permitted Values*	*Default*
voice.vad.signalAnnexB[1]	**0 or 1**	**1**
If 0, there is no change to SDP. If 1, Annex B is used and a new line is added to SDP depending on the setting of `voice.vadEnable`. If `voice.vadEnable` is set to 1, add parameter line `a=fmtp:18 annexb="yes"` below `a=rtpmap…` parameter line (where '18' could be replaced by another payload). If `voice.vadEnable` is set to 0, add parameter line `a=fmtp:18 annexb="no"` below `a=rtpmap…` parameter line (where '18' could be replaced by another payload).		

Parameter	*Permitted Values*	*Default*
voice.vadEnable[1]	**0 or 1**	**0**
If 0, voice activity detection (VAD) is disabled. If 1, VAD is enabled.		
voice.vadThresh[1]	**integer from 0 to 30**	**15**
The threshold for determining what is active voice and what is background noise in dB. Sounds louder than this value will be considered active voice, and sounds quieter than this threshold will be considered background noise. This does not apply to G.729AB codec operation which has its own built-in VAD function.		

[1] Change causes phone to restart or reboot.

<quality monitoring/>

The table Voice Quality Monitoring Parameters shows the Voice Quality Monitoring parameters.

Voice Quality Monitoring Parameters

Parameter	*Permitted Values*	*Default*
voice.qualityMonitoring.collector.alert.moslq.threshold.critical[1]	**0 to 40**	**0**
The threshold value of listening MOS score (MOS-LQ) that causes phone to send a critical alert quality report. Configure the desired MOS value multiplied by 10. If 0 or Null, critical alerts are not generated due to MOS-LQ. For example, a configured value of 28 corresponds to the MOS score 2.8.		
voice.qualityMonitoring.collector.alert.moslq.threshold.warning[1]	**0 to 40**	**0**
Threshold value of listening MOS score (MOS-LQ) that causes phone to send a warning alert quality report. Configure the desired MOS value multiplied by 10. If 0 or Null, warning alerts are not generated due to MOS-LQ. For example, a configured value of 35 corresponds to the MOS score 3.5.		
voice.qualityMonitoring.collector.alert.delay.threshold.critical[1]	**0 to 2000**	**0**
Threshold value of one way delay (in ms) that causes phone to send a critical alert quality report. If 0 or Null, critical alerts are not generated due to one-way delay. One-way delay includes both network delay and end system delay.		
voice.qualityMonitoring.collector.alert.delay.threshold.warning[1]	**0 to 2000**	**0**
Threshold value of one way delay (in ms) that causes phone to send a critical alert quality report. If 0 or Null, warning alerts are not generated due to one-way delay. One-way delay includes both network delay and end system delay.		
voice.qualityMonitoring.collector.enable.periodic[1]	**0 or 1**	**0**
If 0, periodic quality reports are not generated. If 1, periodic quality reports are generated throughout a call.		
voice.qualityMonitoring.collector.enable.session[1]	**0 or1**	**0**
If 0, quality reports are not generated at the end of each call. If 1, reports are generated at the end of each call.		

Parameter	*Permitted Values*	*Default*
voice.qualityMonitoring.collector.enable.triggeredPeriodic[1]	**0 to 2**	**0**
If 0, alert states do not cause periodic reports to be generated. If 1, periodic reports are generated if an alert state is critical. If 2, period reports are generated when an alert state is either warning or critical. Note: This parameter is ignored when `voice.qualityMonitoring.collector.enable.periodic` is 1, since reports are sent throughout the duration of a call.		
voice.qualityMonitoring.collector.period[1]	**5 to 20**	**20**
The time interval between successive periodic quality reports.		
voice.qualityMonitoring.collector.server.x.address[1] **The server address** **voice.qualityMonitoring.collector.server.x.port[1]** **The server port.**	**Dotted-decimal IP address or hostname** **1 to 65535**	**Null** **5060**
The server address and port of a SIP server (report collector) that accepts voice quality reports contained in SIP PUBLISH messages. Set x to 1 as only one report collector is supported at this time.		
voice.qualityMonitoring.rtcpxr.enable[1]	**0 or 1**	**0**
If 0, RTCP-XR packets are not generated. If 1, the packets are generated.		

[1] Change causes phone to restart or reboot.

<rxQoS/>

The following table lists the jitter buffer parameters for wired network interface voice traffic, wireless network interface voice traffic, and push-to-talk interface voice traffic.

Voice Jitter Buffer Parameters

Parameter	*Permitted Values*	*Default*
voice.rxQoS.avgJitter[1] **The typical average jitter.** **voice.rxQoS.maxJitter[1]** **The maximum expected jitter.**	**0 to 80** **0 to 200**	**20** **160**
The average and maximum jitter in milliseconds for wired network interface voice traffic. `avgJitter` – The wired interface minimum depth will be automatically configured to adaptively handle this level of continuous jitter without packet loss. `maxJitter` – The wired interface jitter buffer maximum depth will be automatically configured to handle this level of intermittent jitter without packet loss. Actual jitter above the average but below the maximum may result in delayed audio play out while the jitter buffer adapts, but no packets will be lost. Actual jitter above the maximum value will always result in packet loss. Note that if legacy `voice.audioProfile.x.jitterBuffer.*` parameters are explicitly specified, they will be used to configure the jitter buffer and these `voice.rxQoS` parameters will be ignored.		

Parameter	*Permitted Values*	*Default*
voice.rxQoS.wireless.avgJitter[1] **The typical average jitter.**	**0 to 200**	**70**
voice.rxQoS.wireless.maxJitter[1] **The maximum expected jitter.**	**20 to 500**	**300**
The average and maximum jitter in milliseconds for wireless network interface voice traffic. `avgJitter` – The wireless interface minimum depth will be automatically configured to adaptively handle this level of continuous jitter without packet loss. `maxJitter` – The wireless interface jitter buffer maximum depth will be automatically configured to handle this level of intermittent jitter without packet loss. Actual jitter above the average but below the maximum may result in delayed audio play out while the jitter buffer adapts, but no packets will be lost. Actual jitter above the maximum value will always result in packet loss. Note: if legacy `voice.audioProfile.x.jitterBuffer.*` parameters are explicitly specified, they will be used to configure the jitter buffer and these `voice.rxQoS` parameters will be ignored for wireless interfaces.		
voice.rxQoS.ptt.avgJitter[1] **The typical average jitter.**	**0 to 200**	**150**
voice.rxQoS.ptt.maxJitter[1] **The maximum expected jitter.**	**20 to 500**	**480**
The average and maximum jitter in milliseconds for IP multicast voice traffic (wired or wireless). `avgJitter` – The Paging interface minimum depth will be automatically configured to adaptively handle this level of continuous jitter without packet loss. `maxJitter` – The Paging interface jitter buffer maximum depth will be automatically configured to handle this level of intermittent jitter without packet loss. Actual jitter above the average but below the maximum may result in delayed audio play out while the jitter buffer adapts, but no packets will be lost. Actual jitter above the maximum value will always result in packet loss. Note: if legacy `voice.audioProfile.x.jitterBuffer.*` parameters are explicitly specified, they will be used to configure the jitter buffer and these `voice.rxQoS` parameters will be ignored for Paging interface interfaces.		

[1] Change causes phone to restart or reboot.

<voIpProt/>

You must set up the call server and DTMF signaling parameters.

This parameter includes:

- <server/>
- <SDP/>
- <SIP/>

<server/>

The configuration parameters listed in the table VoIP Server Parameters are defined as follows.

VoIP Server Parameters

Parameter	*Permitted Values*	*Default*
voIpProt.server.dhcp.available[1]	**0 or 1**	**0**
If 0, do not check with the DHCP server for the SIP server IP address. If 1, check with the server for the IP address.		
voIpProt.server.dhcp.option[1]	**128 to 254**	**128**
The option to request from the DHCP server if `voIpProt.server.dhcp.available`= 1. Note: If `reg.x.server.y.address` is non-Null, it takes precedence even if the DHCP server is available.		
voIpProt.server.dhcp.type[1]	**0 or 1**	**0**
Type to request from the DHCP server if `voIpProt.server.dhcp.available` is set to 1.If this parameter is set to 0, IP request address. If set to 1, request string		
voIpProt.server.x.address	**dotted- decimal IP address or hostname**	**Null**
The IP address or hostname and port of a SIP server that accepts registrations. Multiple servers can be listed starting with x=1 to 4 for fault tolerance.		
voIpProt.server.x.port	**0, 1 to 65535**	**0**
The port of the server that specifies registrations. If 0, the port used depends on `voIpProt.server.x.transport`.		
voIpProt.server.x.registerRetry.baseTimeOut	**10 - 120**	**60**
The base time period to wait before a registration retry. Used in conjunction with `voIpProt.server.x.registerRetry.maxTimeOut` to determine how long to wait. The algorithm is defined in RFC 5626. If both parameters `voIpProt.server.x.registerRetry.baseTimeOut` and `reg.x.server.y.registerRetry.baseTimeOut` are set, the value of `reg.x.server.y.registerRetry.baseTimeOut` takes precedence.		
voIpProt.server.x.registerRetry.maxTimeOut	**60 - 1800**	**60**
The maximum time period to wait before a registration retry. Used in conjunction with `voIpProt.server.x.registerRetry.maxTimeOut` to determine how long to wait. The algorithm is defined in RFC 5626. If both parameters `voIpProt.server.x.registerRetry.maxTimeOut` and `reg.x.server.y.registerRetry.maxTimeOut` are set, the value of `reg.x.server.y.registerRetry.maxTimeOut` takes precedence.		

Parameter	*Permitted Values*	*Default*
voIpProt.server.x.transport	**DNSnaptr, TCPpreferred, UDPOnly, TLS, TCPOnly**	**DNSnaptr**
The transport method the phone uses to communicate with the SIP server. `Null` or `DNSnaptr` – if `voIpProt.server.x.address` is a hostname and `voIpProt.server.x.port` is 0 or Null, do NAPTR then SRV look-ups to try to discover the transport, ports and servers, as per RFC 3263. If `voIpProt.server.x.address` is an IP address, or a port is given, then UDP is used. `TCPpreferred` – TCP is the preferred transport; UDP is used if TCP fails. `UDPOnly`**:** only UDP will be used. `TLS` – if TLS fails, transport fails. Leave port field empty (will default to 5061) or set to 5061. `TCPOnly` – only TCP will be used.		
voIpProt.server.x.protocol.SIP	**0 or 1**	**1**
If 1, server is a SIP proxy/registrar. Note: if set to 0, and the server is confirmed to be a SIP server, then the value is assumed to be 1.		
voIpProt.server.x.expires	**positive integer, minimum 10**	**3600**
The phone's requested registration period in seconds. Note: The period negotiated with the server may be different. The phone will attempt to re-register at the beginning of the `overlap` period. For example, if `expires="300"` and `overlap="5"`, the phone will re-register after 295 seconds (300–5).		
voIpProt.server.x.expires.overlap	**5 to 65535**	**60**
The number of seconds before the expiration time returned by server x at which the phone should try to re-register. The phone will try to re-register at half the expiration time returned by the server if the server value is less than the configured overlap value.		
voIpProt.server.x.expires.lineSeize	**positive integer, minimum 0 was 10**	**30**
Requested line-seize subscription period.		
voIpProt.server.x.failOver.failBack.mode	**newRequests, DNSTTL, registration, duration**	**newRequest s**
The mode for failover failback: `newRequests` – all new requests are forwarded first to the primary server regardless of the last used server. `DNSTTL` – the phone tries the primary server again after a timeout equal to the DNS TTL configured for the server that the phone is registered to. `registration` – the phone tries the primary server again when the registration renewal signaling begins. `duration` – the phone tries the primary server again after the time specified by `voIpProt.server.x.failOver.failBack.timeout`.		
voIpProt.server.x.failOver.failBack.timeout	**0, 60 to 65535**	**3600**
If `voIpProt.server.x.failOver.failBack.mode` is set to duration, this is the time in seconds after failing over to the current working server before the primary server is again selected as the first server to forward new requests to. Values between 1 and 59 will result in a timeout of 60 and 0 means do not fail-back until a fail-over event occurs with the current server.		

Parameter	*Permitted Values*	*Default*
voIpProt.server.x.failOver.failRegistrationOn	**0 or 1**	**0**
When set to 1, and the reRegisterOn parameter is enabled, the phone will silently invalidate an existing registration (if it exists), at the point of failing over. When set to 0, and the reRegisterOn parameter is enabled, existing registrations will remain active. This means that the phone will attempt failback without first attempting to register with the primary server to determine if it has recovered.		
voIpProt.server.x.failOver.onlySignalWithRegistered	**0 or 1**	**1**
When set to 1, and the reRegisterOn and failRegistrationOn parameters are enabled, no signaling is accepted from or sent to a server that has failed until failback is attempted or failover occurs. If the phone attempts to send signaling associated with an existing call via an unregistered server (for example, to resume or hold a call), the call will end. No SIP messages will be sent to the unregistered server. When set to 0, and the reRegisterOn and failRegistrationOn parameters are enabled, signaling will be accepted from and sent to a server that has failed (even though failback hasn't been attempted or failover hasn't occurred).		
voIpProt.server.x.failOver.reRegisterOn	**0 or 1**	**0**
When set to 1, the phone will attempt to register with (or via, for the outbound proxy scenario), the secondary server. If the registration succeeds (a 200 OK response with valid expires), signaling will proceed with the secondary server. When set to 0, the phone won't attempt to register with the second.		
voIpProt.server.x.lcs	**0 or 1**	**0**
If 0, the Microsoft Live Communications Server (LSC) is not supported. If 1, LCS is supported for registration x. This parameter overrides `voIpProt.SIP.lcs` .		
voIpProt.server.x.register	**0 or 1**	**1**
If 0, calls can be routed to an outbound proxy without registration. See `reg.x.server.y.register`. For more information, see *Technical Bulletin 5844: SIP Server Fallback Enhancements on Polycom Phones*.		
voIpProt.server.x.retryTimeOut	**0 to 65535**	**0**
The amount of time (in milliseconds) to wait between retries. If 0, use standard RFC 3261 signaling retry behavior.		
voIpProt.server.x.retryMaxCount	**0 to 20**	**3**
If set to 0, 3 is used. The number of retries that will be attempted before moving to the next available server.		
voIpProt.server.x.specialInterop	**standard, ocs2007r2, lcs2005, lync2010**	**standard**
Specify if this registration should support Microsoft Office Communications Server 2007 R2 (ocs2007r2), Microsoft Live Communications Server 2005 (lcs2005), or Microsoft Lync 2010 (lync2010).		
voIpProt.server.x.useOutboundProxy	**0 or 1**	**1**
Specify whether or not to use the outbound proxy specified in `voIpProt.SIP.outboundProxy.address` for server x.		

[1] Change causes phone to restart or reboot.

<SDP/>

The configuration parameters in the table Session Description Protocol Parameters is defined as follows:

Session Description Protocol (SDP) Parameters

Parameter	*Permitted Values*	*Default*
voIpProt.SDP.answer.useLocalPreferences	**0 or 1**	**0**
If set to 1, the phones uses its own preference list when deciding which codec to use rather than the preference list in the offer. If set to 0, it is disabled.		
voIpProt.SDP.early.answerOrOffer	**0 or 1**	**0**
If set to 1, an SDP offer or answer is generated in a provisional reliable response and PRACK request and response. If set to 0, an SDP offer or answer is not generated. Note: An SDP offer or answer is not generated if `reg.x.musicOnHold.uri` is set.		
voIpProt.SDP.iLBC.13_33kbps.includeMode	**0 or 1**	**1**
If set to 1, the phone should include the mode=30 FMTP parameter in SDP offers: If voice.codecPref.iLBC.13_33kbps is set and voice.codecPref.iLBC.15_2kbps is Null. If voice.codecPref.iLBC.13_33kbps and voice.codecPref.iLBC.15_2kbps are both set, the iLBC 13.33 kbps codec is set to a higher preference. If set to 0, the phone should not include the mode=30 FTMP parameter in SDP offers even if iLBC 13.33 kbps codec is being advertised. See <codecPref/>.		
voIpProt.SDP.useLegacyPayloadTypeNegotiation	**0 or 1**	**0**
If set to 1, the phone transmits and receives RTP using the payload type identified by the first codec listed in the SDP of the codec negotiation answer. If set to 0, RFC 3264 is followed for transmit and receive RTP payload type values.		

<SIP/>

The configuration parameters in the table Session Initiation Protocol Parameters is defined as follows.

Session Initiation Protocol (SIP) Parameters

Parameter	*Permitted Values*	*Default*
voIpProt.SIP.acd.signalingMethod[1]	**0 or 1**	**0**
If set to 0, the 'SIP-B' signaling is supported. (This is the older ACD functionality.) If set to 1, the feature synchronization signaling is supported. (This is the new ACD functionality.)		

Parameter	*Permitted Values*	*Default*
voIpProt.SIP.alertInfo.x.class	**see the list of ring classes in <rt/>**	**default**
Alert-Info fields from INVITE requests will be compared against as many of these parameters as are specified (x=1, 2, ..., N) and if a match is found, the behavior described in the corresponding ring class is applied.		
voIpProt.SIP.alertInfo.x.value	**string**	**Null**
A string to match the alertinfo header in the incoming INVITE.		
voIpProt.SIP.allowTransferOnProceeding	**0, 1, 2**	**1**
If set to 0, a transfer is not allowed during the proceeding state of a consultation call. If set to 1, a transfer can be completed during the proceeding state of a consultation call. If set to 2, phones will accept an INVITE with replaces for a dialog in early state. This is needed when using transfer on proceeding with a proxying call server such as openSIPS, reSIProcate or SipXecs.		
voIpProt.SIP.authOptimizedInFailover	**0 or 1**	**0**
If set to 1, when failover occurs, the first new SIP request is sent to the server that sent the proxy authentication request. If set to 0, when failover occurs, the first new SIP request is sent to the server with the highest priority in the server list. If `reg.x.auth.optimizedInFailover` set to 0, this parameter is checked. If `voIpProt.SIP.authOptimizedInFailover` is 0, then this feature is disabled. If both parameters are set, the value of `reg.x.auth.optimizedInFailover` takes precedence.		
voIpProt.SIP.CID.sourcePreference	**ASCII string up to 120 characters long**	**Null**
Specify the priority order for the sources of caller ID information. The headers can be in any order. If Null, caller ID information comes from P-Asserted-Identity, Remote-Party-ID, and From in that order. The values `From,P-Asserted-Identity, Remote-Party-ID` and `P-Asserted-Identity,From, Remote-Party-ID` are also valid.		
voIpProt.SIP.compliance.RFC3261.validate.contentLanguage	**0 or 1**	**1**
If set to 1, validation of the SIP header content language is enabled. If set to 0, validation is disabled.		
voIpProt.SIP.compliance.RFC3261.validate.contentLength	**0 or 1**	**1**
If set to 1, validation of the SIP header content length is enabled.		
voIpProt.SIP.compliance.RFC3261.validate.uriScheme	**0 or 1**	**1**
If set to 1, validation of the SIP header URI scheme is enabled. If set to 0, validation is disabled.		
voIpProt.SIP.conference.address	**ASCII string up to 128 characters long**	**Null**
If Null, conferences are set up on the phone locally. If set to some value, conferences are set up by the server using the conferencing agent specified by this address. Acceptable values depend on the conferencing server implementation policy.		

Parameter	*Permitted Values*	*Default*
voIpProt.SIP.conference.parallelRefer	**0 or 1**	**0**
If 1, a parallel REFER is sent to the call server. **Note**: This parameter must be set for Siemens Openscape Centralized Conferencing.		
voIpProt.SIP.connectionReuse.useAlias	**0 or 1**	**0**
If set to 0, the alias parameter is not added to the via header If set to 1, the phone uses the connection reuse draft which introduces "alias".		
voIpProt.SIP.csta	**0 or 1**	**0**
If 0, the uaCSTA (User Agent Computer Supported Telecommunications Applications) feature is disabled. If 1, uaCSTA is enabled (If `reg.x.csta` is set, it will override this parameter).		
voIpProt.SIP.dialog.strictXLineID	**0 or 1**	**0**
If 0, the phone will not look for x-line-id (call appearance indec) in a SIP INVITE message, if one is not present. Instead, when it receives INVITE, the phone will generate the call appearance locally and pass that information to other parties involved in the call.		
voIpProt.SIP.dialog.usePvalue	**0 or 1**	**0**
If set to 0, phone uses a `pval` field name in Dialog. This obeys the draft-ietf-sipping-dialog-package-06.txt draft. If set to 1, the phone uses a field name of `pvalue`.		
voIpProt.SIP.dialog.useSDP	**0 or 1**	**0**
If set to 0, a new dialog event package draft is used (no SDP in dialog body). If set to 1, for backwards compatibility, use this setting to send SDP in the dialog body.		
voIpProt.SIP.dtmfViaSignaling.rfc2976[1]	**0 or 1**	**0**
If set to 1, DTMF digit information is sent in RFC2976 SIP INFO packets during a call. If set to 0, no DTMF digit information is sent.		
voIpProt.SIP.enable[1]	**0 or 1**	**1**
A flag to determine if the SIP protocol is used for call routing, dial plan, DTMF, and URL dialing. If set to 1, the SIP protocol is used.		
voIpProt.SIP.failoverOn503Response	**0 or 1**	**1**
A flag to determine whether or not to trigger a failover if the phone receives a 503 response.		
voIpProt.SIP.header.diversion.enable[1]	**0 or 1**	**0**
If set to 1, the diversion header is displayed if received. If set to 0, the diversion header is not displayed.		
voIpProt.SIP.header.diversion.list.useFirst[1]	**0 or 1**	**1**
If set to 1, the first diversion header is displayed. If set to 0, the last diversion header is displayed.		

Parameter	*Permitted Values*	*Default*
voIpProt.SIP.header.warning.codes.accept	**comma separated list**	**Null**
Specify a list of accepted warning codes. If set to Null, all codes are accepted. Only codes between 300 and 399 are supported. For example, if you want to accept only codes 325 to 330: `voIpProt.SIP.header.warning.codes.accept=325,326,327,328,329,330` Text will be shown in the appropriate language. For more information, see lcl_ml_lang_menu_x.		
voIpProt.SIP.header.warning.enable	**0 or 1**	**0**
If set to 1, the warning header is displayed if received. If set to 0, the warning header is not displayed.		
voIpProt.SIP.IM.autoAnswerDelay	**0 to 40, seconds**	**10**
The time interval from receipt of the instant message invitation to automatically accepting the invitation.		
voIpProt.SIP.keepalive.sessionTimers	**0 or 1**	**0**
If set to 1, the session timer will be enabled. If set to 0, the session timer will be disabled, and the phone will not declare "timer" in "Support" header in an INVITE. The phone will still respond to a re-INVITE or UPDATE. The phone will not try to re-INVITE or UPDATE even if the remote endpoint asks for it.		
voIpProt.SIP.lcs	**0 or 1**	**0**
If 0, the Microsoft Live Communications Server (LCS) is not supported. If 1, LCS is supported. This parameter can set for a specific registration using `reg.x.lcs`.		
voIpProt.SIP.lineSeize.retries	**3 to 10**	**10**
Controls the number of times the phone will retry a notify when attempting to seize a line (BLA).		
voIpProt.SIP.local.port[1]	**0 to 65535**	**5060**
The local port for sending and receiving SIP signaling packets. If set to 0, 5060 is used for the local port but is not advertised in the SIP signaling. If set to some other value, that value is used for the local port and it is advertised in the SIP signaling.		
voIpProt.SIP.ms-forking	**0 or 1**	**0**
If set to 0, support for MS-forking is disabled. If set to 1, support for MS-forking is enabled and the phone will reject all Instant Message INVITEs. This parameter is applies when installing Microsoft Live Communications Server. Note that if any endpoint registered to the same account has MS-forking disabled, all other endpoints default back to non-forking mode. Windows Messenger does not use MS-forking so be aware of this behavior if one of the endpoints is using Windows Messenger.		
voIpProt.SIP.mtls.enable	**0 or 1**	**1**
If 0, Mutual TLS is disabled. If 1, Mutual TLS is enabled. Used in conjunction with Microsoft Lync 2010.		
voIpProt.SIP.musicOnHold.uri	**a SIP URI**	**Null**
A URI that provides the media stream to play for the remote party on hold. This parameter is used if `reg.x.musicOnHold.uri` is Null. Note: The SIP URI parameter transport is supported when configured with the values of `UDP`, `TCP`, or `TLS`.		

Parameter	*Permitted Values*	*Default*
voIpProt.SIP.outboundProxy.address	**dotted-decimal IP address or hostname**	**Null**
The IP address or hostname of the SIP server to which the phone sends all requests.		
voIpProt.SIP.outboundProxy.port	**0 to 65535**	**0**
The port of the SIP server to which the phone sends all requests.		
voIpProt.SIP.outboundProxy.failOver.failBack.mode	**newRequests, DNSTTL, registration, duration,**	**newRequests**
The mode for failover failback (overrides `voIpProt.server.x.failOver.failBack.mode`). `newRequests` – all new requests are forwarded first to the primary server regardless of the last used server. `DNSTTL` – the phone tries the primary server again after a timeout equal to the DNS TTL configured for the server that the phone is registered to. `registration` – the phone tries the primary server again when the registration renewal signaling begins. `duration` – the phone tries the primary server again after the time specified by `reg.x.outboundProxy.failOver.failBack.timeout` expires.		
voIpProt.SIP.outboundProxy.failOver.failBack.timeout	**0, 60 to 65535**	**3600**
The time to wait (in seconds) before failback occurs (overrides `voIpProt.server.x.failOver.failBack.timeout`).If the fail back mode is set to Duration, the phone waits this long after connecting to the current working server before selecting the primary server again. If 0, the phone will not fail-back until a fail-over event occurs with the current server.		
voIpProt.SIP.outboundProxy.failOver.failRegistrationOn	**0 or 1**	**0**
When set to 1, and the reRegisterOn parameter is enabled, the phone will silently invalidate an existing registration (if it exists), at the point of failing over. When set to 0, and the reRegisterOn parameter is enabled, existing registrations will remain active. This means that the phone will attempt failback without first attempting to register with the primary server to determine if it has recovered. Note that `voIpProt.SIP.outboundProxy.failOver.RegisterOn` must be enabled.		
voIpProt.SIP.outboundProxy.failOver.onlySignalWithRegistered	**0 or 1**	**1**
When set to 1, and the reRegisterOn and failRegistrationOn parameters are enabled, no signaling is accepted from or sent to a server that has failed until failback is attempted or failover occurs. If the phone attempts to send signaling associated with an existing call via an unregistered server (for example, to resume or hold a call), the call will end. No SIP messages will be sent to the unregistered server. When set to 0, and the reRegisterOn and failRegistrationOn parameters are enabled, signaling will be accepted from and sent to a server that has failed (even though failback hasn't been attempted or failover hasn't occurred). This parameter overrides voIpProt.server.x.failOver.onlySignalWithRegistered.		
voIpProt.SIP.outboundProxy.failOver.reRegisterOn	**0 or 1**	**0**
This parameter overrides the `voIpProt.server.x.failOver.reRegisterOn`. When set to 1, the phone will attempt to register with (or via, for the outbound proxy scenario), the secondary server. If the registration succeeds (a 200 OK response with valid expires), signaling will proceed with the secondary server. When set to 0, the phone won't attempt to register with the secondary server, since the phone will assume that the primary and secondary servers share registration information.		

Parameter	*Permitted Values*	*Default*
voIpProt.SIP.outboundProxy.transport	**DNSnaptr, TCPpreferred, UDPOnly, TLS, TCPOnly**	**DNSnaptr**
The transport method the phone uses to communicate with the SIP server. `Null` or `DNSnaptr` – if `reg.x.outboundProxy.address` is a hostname and `reg.x.outboundProxy.port` is 0 or Null, do NAPTR then SRV look-ups to try to discover the transport, ports and servers, as per RFC 3263. If `reg.x.outboundProxy.address` is an IP address, or a port is given, then UDP is used. `TCPpreferred` – TCP is the preferred transport, UDP is used if TCP fails. `UDPOnly` – only UDP will be used. `TLS` – if TLS fails, transport fails. Leave port field empty (will default to 5061) or set to 5061. `TCPOnly` – only TCP will be used.		
voIpProt.SIP.pingInterval	**0 to 3600**	**0**
The number in seconds to send "PING" message. This feature is disabled by default.		
voIpProt.SIP.pingMethod	**PING, OPTIONS**	**PING**
The ping method to be used.		
voIpProt.SIP.presence.nortelShortMode[1]	**0 or 1**	**0**
Different headers sent in SUBSCRIBE when used for presence on an Avaya (Nortel) server. Support is indicated by adding a header `Accept-Encoding: x-nortel-short`. A PUBLISH is sent to indicate the status of the phone.		
voIpProt.SIP.requestValidation.digest.realm[1]	**A valid string**	**PolycomSPIP**
Determines the string used for Realm.		
voIpProt.SIP.requestValidation.x.method[1]	**Null, source, digest, both, all**	**Null**
If Null, no validation is made. Otherwise this sets the type of validation performed for the request: source: ensure request is received from an IP address of a server belonging to the set of target registration servers; digest: challenge requests with digest authentication using the local credentials for the associated registration (line); both or all: apply both of the above methods		
voIpProt.SIP.requestValidation.x.request[1]	**INVITE, ACK , BYE, REGISTER, CANCEL, OPTIONS, INFO, MESSAGE, SUBSCRIBE, NOTIFY, REFER, PRACK, UPDATE**	**Null**
Sets the name of the method for which validation will be applied. Note: Intensive request validation may have a negative performance impact due to the additional signaling required in some cases.		

Parameter	*Permitted Values*	*Default*
voIpProt.SIP.requestValidation.x.request.y.event[1]	**A valid string**	**Null**
Determines which events specified with the Event header should be validated; only applicable when `voIpProt.SIP.requestValidation.x.request` is set to `SUBSCRIBE` or `NOTIFY`. If set to Null, all events will be validated.		
voIpProt.SIP.requestURI.E164.addGlobalPrefix	**0 or 1**	**0**
If set to 1, ‘+’ global prefix is added to the E.164 user parts in sip: URIs.		
voIpProt.SIP.sendCompactHdrs	**0 or 1**	**0**
If set to 0, SIP header names generated by the phone use the long form, for example `From`. If set to 1, SIP header names generated by the phone use the short form, for example `f`.		
voIpProt.SIP.serverFeatureControl.cf[1]	**0 or 1**	**0**
If set to 1, server-based call forwarding is enabled. The call server has control of call forwarding. If set to 0, server-based call forwarding is not enabled.		
voIpProt.SIP.serverFeatureControl.dnd[1]	**0 or 1**	**0**
If set to 1, server-based DND is enabled. The call server has control of DND. If set to 0, server-based DND is not enabled.		
voIpProt.SIP.serverFeatureControl.missedCalls[1]	**0 or 1**	**0**
If set to 1, server-based missed calls is enabled. The call server has control of missed calls. If set to 0, server-based missed calls is not enabled.		
voIpProt.SIP.serverFeatureControl.localProcessing.cf	**0 or 1**	**1**
If set to 0 and `voIpProt.SIP.serverFeatureControl.cf` is set to 1, the phone will not perform local Call Forward behavior. If set to 1, the phone will perform local Call Forward behavior on all calls received.		
voIpProt.SIP.serverFeatureControl.localProcessing.dnd	**0 or 1**	**1**
If set to 0 and `voIpProt.SIP.serverFeatureControl.dnd` is set to 1, the phone will not perform local DND call behavior. If set to 1, the phone will perform local DND call behavior on all calls received.		
voIpProt.SIP.specialEvent.checkSync.alwaysReboot[1]	**0 or 1**	**0**
If set to 1, always reboot when a NOTIFY message is received from the server with event equal to check-sync. If set to 0, only reboot if any of the files listed in **<MAC-address>.cfg** have changed on the FTP server when a NOTIFY message is received from the server with event equal to check-sync.		
voIpProt.SIP.specialEvent.lineSeize.nonStandard[1]	**0 or 1**	**1**
If set to 1, process a 200 OK response for a line-seize event SUBSCRIBE as though a line-seize NOTIFY with Subscription State: active header had been received,. This speeds up processing.		

Parameter	*Permitted Values*	*Default*
voIpProt.SIP.strictLineSeize	**0 or 1**	**0**
If set to 1, The phone is forced to wait for a 200 OK response when receiving a TRYING notify. If set to 0, dial prompt is provided immediately when you attempt to seize a shared line without waiting for a successful OK from the call server.		
voIpProt.SIP.strictUserValidation	**0 or 1**	**0**
If set to 1, the phone is forced to match the user portion of signaling exactly. If set to 0, the phone will use the first registration if the user part does not match any registration.		
voIpProt.SIP. supportFor100rel	**0 or 1**	**1**
If set to 1, the phone advertises support for reliable provisional responses in its offers and responses. If set to 0, the phone will not offer 100rel and will reject offers requiring 100rel.		
voIpProt.SIP.tcpFastFailover	**0 or 1**	**0**
If set to 1, failover occurs based on the values of `reg.x.server.y.retryMaxCount` and `voIpProt.server.x.retryTimeOut`. If 0, a full 32 second RFC compliant timeout is used. See `reg.x.tcpFastFailover` .		
voIpProt.SIP.tlsDsk.enable	**0 or 1**	**0**
If 0, TLS DSK is disabled. If 1, TLS DSK is enabled. For more information, see Session Initiation Protocol (SIP) Authentication Extensions Protocol Overview.		
voIpProt.SIP.turnOffNonSecureTransport[1]	**0 or 1**	**0**
If set to 1, stop listening to port 5060 when using AS-SIP enabled.		
voIpProt.SIP.use486forReject	**0 or 1**	**0**
If set to 1 and the phone is indicating a ringing inbound call appearance, the phone will transmit a 486 response to the received INVITE when the Reject soft key is pressed. If set to 0, no 486 response is transmitted.		
voIpProt.SIP.useContactInReferTo	**0 or 1**	**0**
If set to 0, the "To URI" is used in the REFER. If set to 1, the "Contact URI" is used in the REFER.		
voIpProt.SIP.useRFC2543hold	**0 or 1**	**0**
If set to 0, use SDP media direction parameters (such as a=sendonly) per RFC 3264 when initiating a call. Otherwise use the obsolete c=0.0.0.0 RFC2543 technique. In either case, the phone processes incoming hold signaling in either format. Note: voIpProt.SIP.useRFC2543hold is effective only when the call is initiated.		
voIpProt.SIP.useSendonlyHold	**0 or 1**	**1**
If set to 1, the phone will send a reinvite with a stream mode parameter of "sendonly" when a call is put on hold. This is the same as the previous behavior. If set to 0, the phone will send a reinvite with a stream mode parameter of "inactive" when a call is put on hold. NOTE: The phone will ignore the value of this parameter if set to 1 when the parameter voIpProt.SIP.useRFC2543hold is also set to 1 (default is 0).		

Parameter	*Permitted Values*	*Default*

[1] Change causes phone to restart or reboot.

<webutility/>

The parameters listed in the table Web Configuration Utility Parameters specify the download location of the translated language files for the Web Configuration Utility.

Web Configuration Utility Parameters

Parameter	*Permitted Values*	*Default*
webutility.langauge.plcm.server.url	**URL**	**http://downloads.polycom.com/voice/software/languages/**
The download location of the translated language files for the Web Configuration Utility.		

<xmpp/>

The parameters in the table XML Streaming Protocol Parameters set the XML streaming protocols for instant messaging, presence, and contact list maintance for BroadSoft features

XML Streaming Protocol Parameters

Parameter	*Permitted Values*	*Default*
xmpp.1.auth.password	**UTF-8 encoded string**	**Null**
Password used for XMPP registration. When provisioned from CMA, the value is set to the CMA account password.		
xmpp.1.dialMethod	**String min 0, max 256**	**SIP**
For SIP dialing, the destination XMPP URI is converted to a SIP URI, and the first available SIP line is used to place the call.		
xmpp.1.enable	**0 or 1**	**0**
Flag to determine if XMPP presence is enabled . If 1 XMPP presence is enabled.		
xmpp.1.jid	**String min 0, max 256**	**Null**
Jabber identity used to register with presence server . For example: `presence.test2@polycom-alpha.eu.bc.im`.		

Parameter	*Permitted Values*	*Default*
xmpp.1.roster.invite.accept	**Automatic or prompt**	**Prompt**
Turns the BroadSoft XMPP inviter's subscription for presence. If set to prompt, phone receives pending invitation successfully and can accept or reject the invitation.		
xmpp.1.server	**dotted-decimal IP address, host name, or FQDN**	**Null**
Sets the BroadSoft XMPP presence server to IP or FQDN. For example: `polycom-alpha.eu.bc.im`.		
xmpp.1.verifyCert	**0 or 1**	**1**
Enables and disables the Server Certificate verification from XMPP server. Accepted Values: 0 – Disables; 1 – Enables. If 0, verification of the TLS certificate provided by the BroadSoft XMPP presence server is turned off.		

Session Initiation Protocol (SIP)

This section describes the basic Session Initiation Protocol (SIP) and the protocol extensions that the current Polycom UC Software supports.

This section contains the following information:

- Basic Protocols   All the basic calling functionality described in the SIP specification is supported. Transfer is included in the basic SIP support.
- Protocol Extensions   Extensions add features to SIP that are applicable to a range of applications, including reliable 1xx responses and session timers.

For information on supported RFCs and Internet drafts, see the section RFC and Internet Draft Support.

You can find information on the following topics:

- Request Support
- Header Support
- Response Support
- Hold Implementation
- Reliability of Provisional Responses
- Transfer
- Third Party Call Control
- SIP for Instant Messaging and Presence Leveraging Extensions
- Shared Call Appearance Signaling
- Bridged Line Appearance Signaling

RFC and Internet Draft Support

The following RFC's and Internet drafts are supported. For more information on any of the documents, enter the RFC number at Request for Comments (RFC).

- RFC 1321—The MD5 Message-Digest Algorithm
- RFC 2327—SDP: Session Description Protocol
- RFC 2387—The MIME Multipart / Related Content-type
- RFC 2976—The SIP INFO Method
- RFC 3261—SIP: Session Initiation Protocol (replacement for RFC 2543)
- RFC 3262—Reliability of Provisional Responses in the Session Initiation Protocol (SIP)
- RFC 3263—Session Initiation Protocol (SIP): Locating SIP Servers
- RFC 3264—An Offer / Answer Model with the Session Description Protocol (SDP)
- RFC 3265—Session Initiation Protocol (SIP) - Specific Event Notification
- RFC 3311—The Session Initiation Protocol (SIP) UPDATE Method
- RFC 3325—SIP Asserted Identity

- RFC 3420—Internet Media Type message/sipfrag
- RFC 3515—The Session Initiation Protocol (SIP) Refer Method
- RFC 3555—MIME Type of RTP Payload Formats
- RFC 3611—RTP Control Protocol Extended reports (RTCP XR)
- RFC 3665—Session Initiation Protocol (SIP) Basic Call Flow Examples
- draft-ietf-sip-cc-transfer-05.txt—SIP Call Control - Transfer
- RFC 3725—Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)
- RFC 3842—A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP)
- RFC 3856—A Presence Event Package for Session Initiation Protocol (SIP)
- RFC 3891—The Session Initiation Protocol (SIP) "Replaces" Header
- RFC 3892—The Session Initiation Protocol (SIP) Referred-By Mechanism
- RFC 3959—The Early Session Disposition Type for the Session Initiation Protocol (SIP)
- RFC 3960—Early Media and Ringing Tone Generation in the Session Initiation Protocol (SIP)
- RFC 3968—The Internet Assigned Number Authority (IANA) Header Field Parameter Registry for the Session Initiation Protocol (SIP)
- RFC 3969—The Internet Assigned Number Authority (IANA) Uniform Resource Identifier (URI) Parameter Registry for the Session Initiation Protocol (SIP)
- RFC 4028—Session Timers in the Session Initiation Protocol (SIP)
- RFC 4235—An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP)
- draft-levy-sip-diversion-08.txt—Diversion Indication in SIP
- draft-anil-sipping-bla-02.txt—Implementing Bridged Line Appearances (BLA) Using Session Initiation Protocol (SIP)
- draft-ietf-sip-privacy-04.txt—SIP Extensions for Network-Asserted Caller Identity and Privacy within Trusted Networks
- draft-ietf-sipping-cc-conferencing-03.txt—SIP Call Control - Conferencing for User Agents
- draft-ietf-sipping-rtcp-summary-02.txt —Session Initiation Protocol Package for Voice Quality Reporting Event
- draft-ietf-sip-connect-reuse-04.txt—Connection Reuse in the Session Initiation Protocol (SIP)

Request Support

The SIP request messages in the table Supported SIP Request Messages are supported:

Supported SIP Request Messages

Method	*Supported*	*Notes*
REGISTER	Yes	
INVITE	Yes	

Method	*Supported*	*Notes*
ACK	Yes	
CANCEL	Yes	
BYE	Yes	
OPTIONS	Yes	
SUBSCRIBE	Yes	
NOTIFY	Yes	
REFER	Yes	
PRACK	Yes	
INFO	Yes	RFC 2976, the phone does not generate INFO requests, but will issue a final response upon receipt. No INFO message bodies are parsed.
MESSAGE	Yes	Final response is sent upon receipt. Message bodies of type text/plain are sent and received.
UPDATE	Yes	

Header Support

The table Supported SIP Request Headers lists the SIP request headers supported.

Note: Reading the Following Table

In the following table, a *Yes* in the Supported column means the header is sent and properly parsed.

Supported SIP Request Headers

Header	*Supported*
Accept	Yes
Accept-Encoding	Yes
Accept-Language	Yes
Accept-Resource-Priority	Yes
Access-Network-Info	No

Header	*Supported*
Access-URL	Yes
Alert-Info	Yes
Allow	Yes
Allow-Events	Yes
Authentication-Info	Yes
Authorization	Yes
Call-ID	Yes
Call-Info	Yes
Contact	Yes
Content-Disposition	Yes
Content-Encoding	Yes
Content-Language	Yes
Content-Length	Yes
Content-Type	Yes
CSeq	Yes
Date	Yes (for missed call, not used to adjust the time of the phone)
Diversion	Yes
Error-Info	No
Event	Yes
Expires	Yes
Flow-Timer	Yes
From	Yes
In-Reply-To	No
Join	Yes
Max-Forwards	Yes
Min-Expires	Yes
Min-SE	Yes
MIME-Version	No

Header	*Supported*
Missed-Calls	Yes
ms-client-diagnostics	Yes
ms-keep-alive	Yes
ms-text-format	Yes
Organization	No
P-Asserted-Identity	Yes
P-Preferred-Identity	Yes
Priority	No
Privacy	No
Proxy-Authenticate	Yes
Proxy-Authorization	Yes
Proxy-Require	Yes
RAck	Yes
Reason	Yes
Record-Route	Yes
Refer-Sub	Yes
Refer-To	Yes
Referred-By	Yes
Referred-To	Yes
Remote-Party-ID	Yes
Replaces	Yes
Reply-To	No
Requested-By	No
Require	Yes
Resource-Priority	Yes
Response-Key	No
Retry-After	Yes
Route	Yes

Header	*Supported*
RSeq	Yes
Server	Yes
Session-Expires	Yes
SIP-Etag	Yes
SIP-If-Match	Yes
Subject	Yes
Subscription-State	Yes
Supported	Yes
Timestamp	Yes
To	Yes
Unsupported	Yes
User-Agent	Yes
Via	Yes
voice-missed-call	Yes
Warning	Yes (Only warning codes 300 to 399)
WWW-Authenticate	Yes
X-Sipx-Authidentity	Yes

Response Support

The SIP responses are listed in the following tables:

- Supported 1xx SIP Responses
- Supported 2xx SIP Responses
- Supported 3xx SIP Responses
- Supported 4xx SIP Responses
- Supported 5xx SIP Responses
- Supported 6xx SIP Responses

Note: Reading the Following Tables

In the following table, a *Yes* in the Supported column means the header is sent and properly parsed. The phone may not actually generate the response.

1xx Responses - Provisional

Supported 1xx SIP Responses

Response	*Supported*
100 Trying	Yes
180 Ringing	Yes
181 Call Is Being Forwarded	No
182 Queued	No
183 Session Progress	Yes

2xx Responses - Success

Supported 2xx SIP Responses

Response	*Supported*	*Notes*
200 OK	Yes	
202 Accepted	Yes	In REFER transfer.

3xx Responses - Redirection

Supported 3xx SIP Responses

Response	*Supported*
300 Multiple Choices	Yes
301 Moved Permanently	Yes

Response	*Supported*
302 Moved Temporarily	Yes
305 Use Proxy	No
380 Alternative Service	No

4xx Responses - Request Failure

Note: Handling 4xx Responses

All 4xx responses for which the phone does not provide specific support will be treated the same as 400 Bad Request.

Supported 4xx SIP Responses

Response	*Supported*
400 Bad Request	Yes
401 Unauthorized	Yes
402 Payment Required	No
403 Forbidden	No
404 Not Found	Yes
405 Method Not Allowed	Yes
406 Not Acceptable	No
407 Proxy Authentication Required	Yes
408 Request Timeout	No
410 Gone	No
413 Request Entity Too Large	No
414 Request-URI Too Long	No
415 Unsupported Media Type	Yes
416 Unsupported URI Scheme	No
420 Bad Extension	No

Response	*Supported*
421 Extension Required	No
423 Interval Too Brief	Yes
480 Temporarily Unavailable	Yes
481 Call/Transaction Does Not Exist	Yes
482 Loop Detected	Yes
483 Too Many Hops	No
484 Address Incomplete	Yes
485 Ambiguous	No
486 Busy Here	Yes
487 Request Terminated	Yes
488 Not Acceptable Here	Yes
491 Request Pending	No
493 Undecipherable	No

5xx Responses - Server Failure

Supported 5xx SIP Responses

Response	*Supported*
500 Server Internal Error	Yes
501 Not Implemented	Yes
502 Bad Gateway	No
503 Service Unavailable	No
504 Server Time-out	No
505 Version Not Supported	No
513 Message Too Large	No

6xx Responses - Global Failure

Supported 6xx SIP Responses

Response	*Supported*
600 Busy Everywhere	No
603 Decline	Yes
604 Does Not Exist Anywhere	No
606 Not Acceptable	No

Hold Implementation

The phone supports two currently accepted means of signaling hold.

The first method, no longer recommended due in part to the RTCP problems associated with it, is to set the "c" destination addresses for the media streams in the SDP to zero, for example, c=0.0.0.0.

The second, and preferred, method is to signal the media directions with the "a" SDP media attributes sendonly, recvonly, inactive, or sendrecv. The hold signaling method used by the phone is configurable (see SIP), but both methods are supported when signaled by the remote endpoint

Note: Hold Methods

Even if the phone is set to use c=0.0.0.0, it will not do so if it gets any sendrecv, sendonly, or inactive from the server. These flags will cause it to revert to the other hold method.

Reliability of Provisional Responses

The phone fully supports RFC 3262 - Reliability of Provisional Responses.

Transfer

The phone supports transfer using the REFER method specified in draft-ietf-sip-cc-transfer-05 and RFC 3515.

Third Party Call Control

The phone supports the delayed media negotiations (INVITE without SDP) associated with third-party call-control applications.

When used with an appropriate server, the User Agent Computer Supported Telecommunications Applications (uaCSTA) feature on the phone may be used for remote control of the phone from computer applications such as Microsoft Office Communicator.

The phone is compliant with "Using CSTA for SIP Phone User Agents (uaCSTA), ECMA TR/087" for the Answer Call, Hold Call, and Retrieve Call functions and "Services for Computer Supported Telecommunications Applications Phase III, ECMA – 269" for the Conference Call function.

This feature is enabled by configuration parameters described in <SIP/> and <reg/> and needs to be activated by a feature application key.

SIP for Instant Messaging and Presence Leveraging Extensions

The phone is compatible with the Presence and Instant Messaging features of Microsoft Windows Messenger 5.1. In a future release, support for the Presence and Instant Message recommendations in the SIP Instant Messaging and Presence Leveraging Extensions (SIMPLE) proposals will be provided by the following Internet drafts or their successors:

- draft-ietf-simple-cpim-mapping-01
- draft-ietf-simple-presence-07
- draft-ietf-simple-presencelist-package-00
- draft-ietf-simple-winfo-format-02
- draft-ietf-simple-winfo-package-02

Shared Call Appearance Signaling

A shared line is an address of record managed by a call server. The server allows multiple endpoints to register locations against the address of record.

The phone supports shared call appearances (SCA) using the SUBSCRIBE-NOTIFY method in the "SIP Specific Event Notification" framework (RFC 3265). The events used are:

- *call-info* for call appearance state notification
- *line-seize* for the phone to ask to seize the line

Bridged Line Appearance Signaling

A bridged line is an address of record managed by a server. The server allows multiple endpoints to register locations against the address of record.

The phone supports bridged line appearances (BLA) using the SUBSCRIBE-NOTIFY method in the "SIP Specific Event Notification" framework (RFC 3265). The events used are:

- "dialog" for bridged line appearance subscribe and notify.

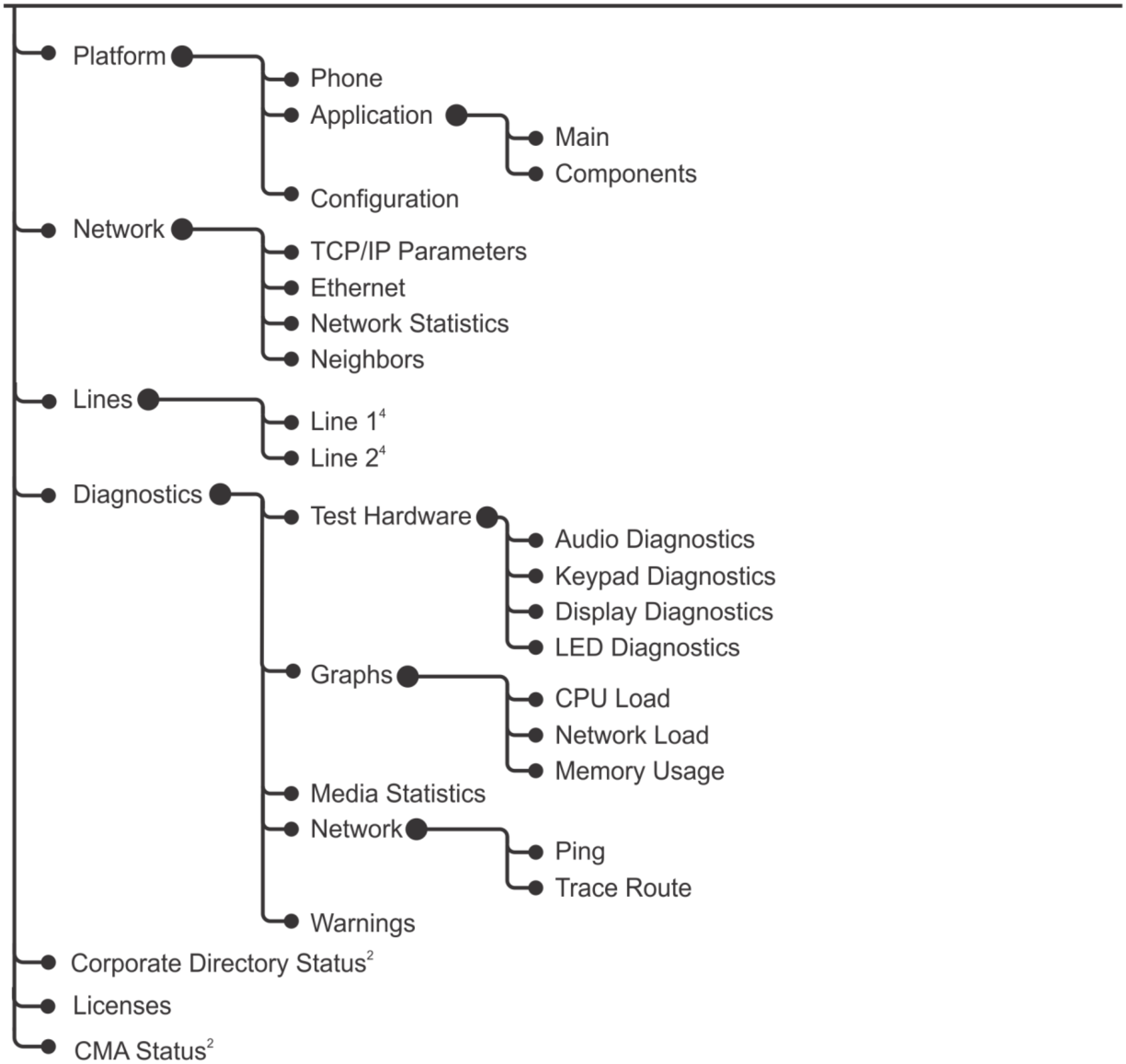
Polycom UC Software Menu System

Features

- Do Not Disturb[1]
- Forward
  - Always
  - No Answer
  - Busy
- Contact Directory
- Corporate Directory[2]
- CMA Directory[2]
- Call Lists
  - Missed Calls
  - Received Calls
  - Placed Calls
  - Clear
- Presence[2]
- Messages[3]
- User Login[2]
  - Login
- Paging/PTT Configuration
  - Paging Groups List
  - PTT Channels List
- Removable Storage Media[2,3]
  - Browse Recordings
  - Picture Frame
  - Storage Media Properties

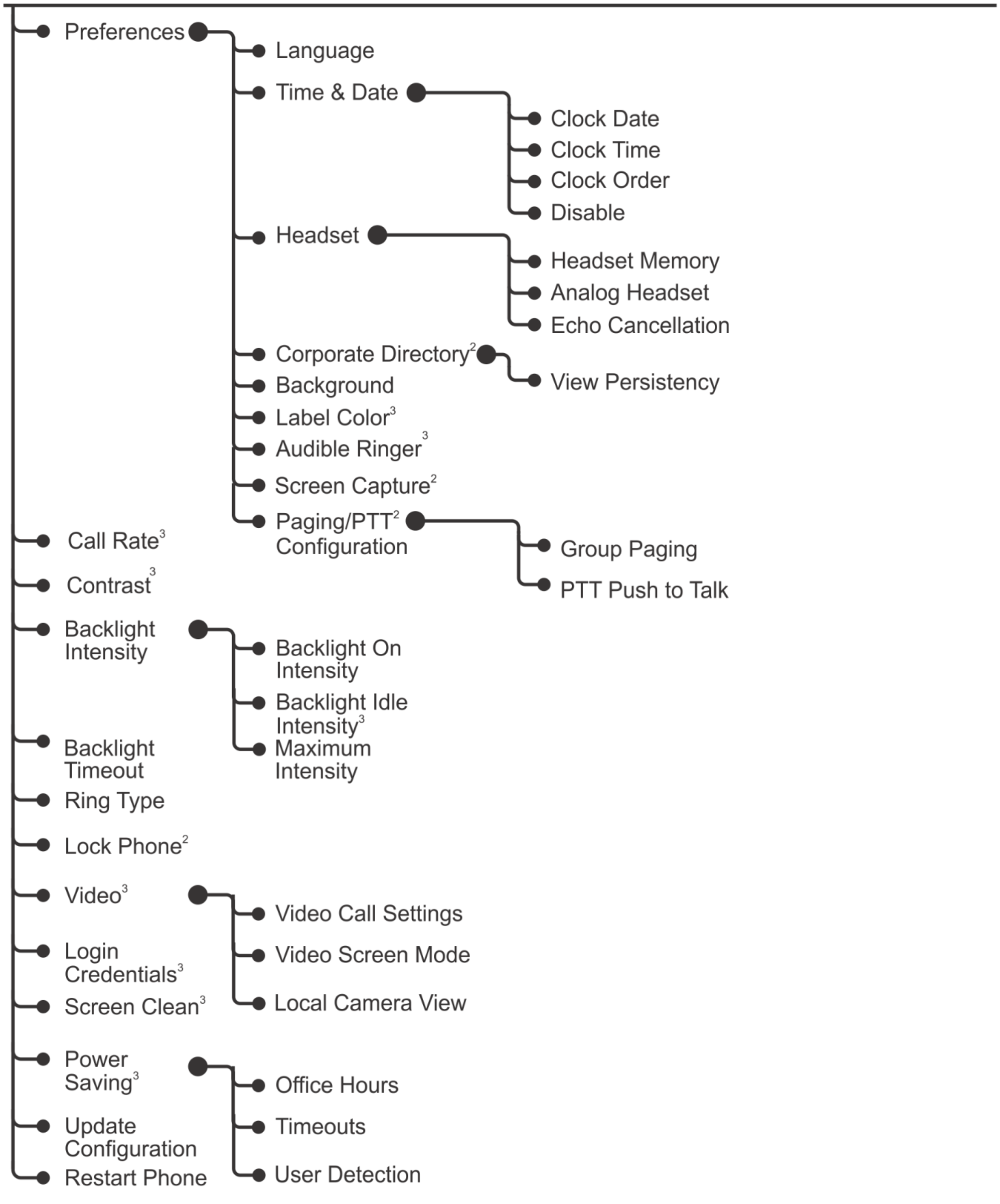
[1] *If no hard key available.*
[2] *If enabled.*
[3] *Platform dependent.*

Status



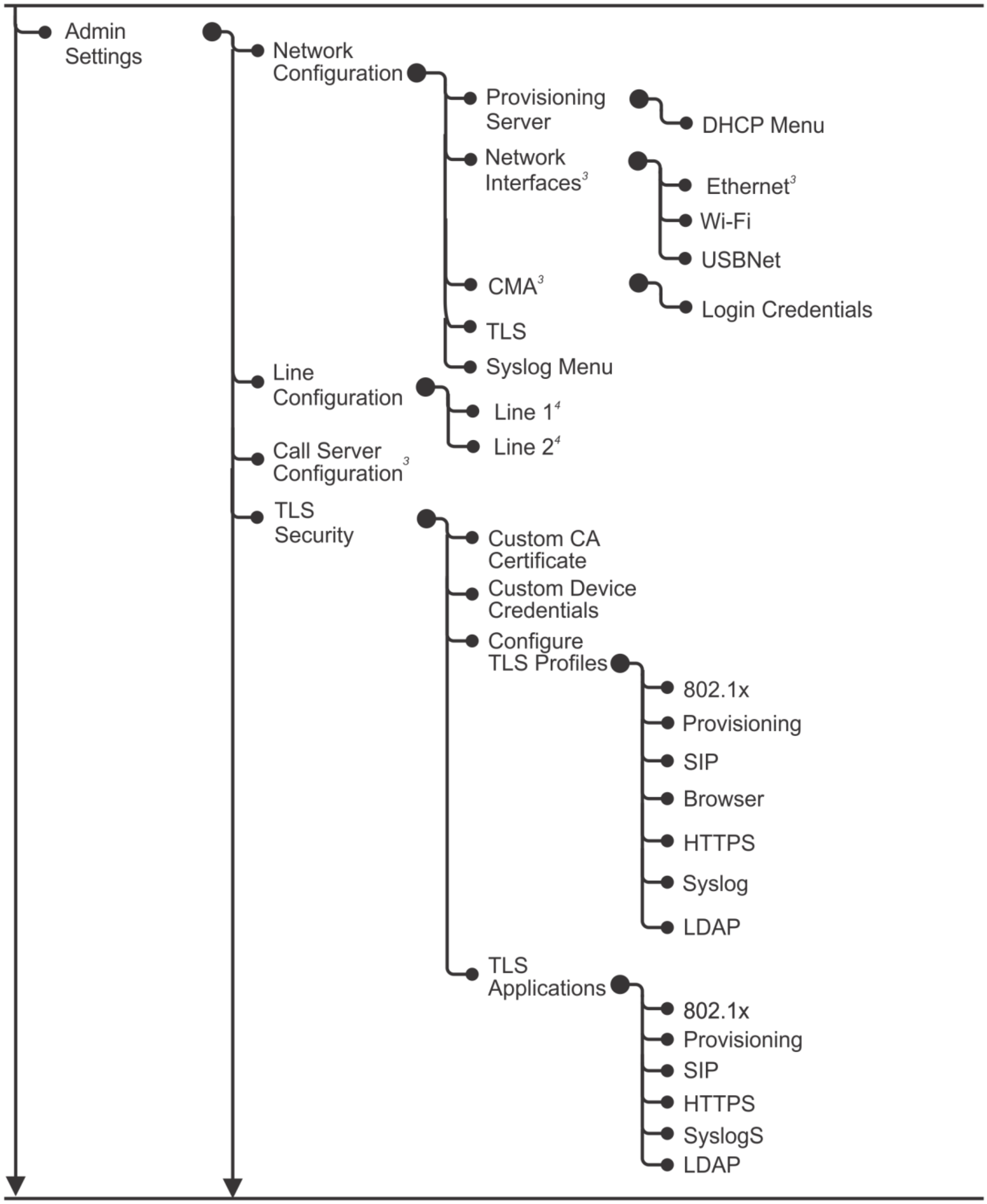
[2] *If enabled.*
[4] *If applicable.*

Settings > Basic



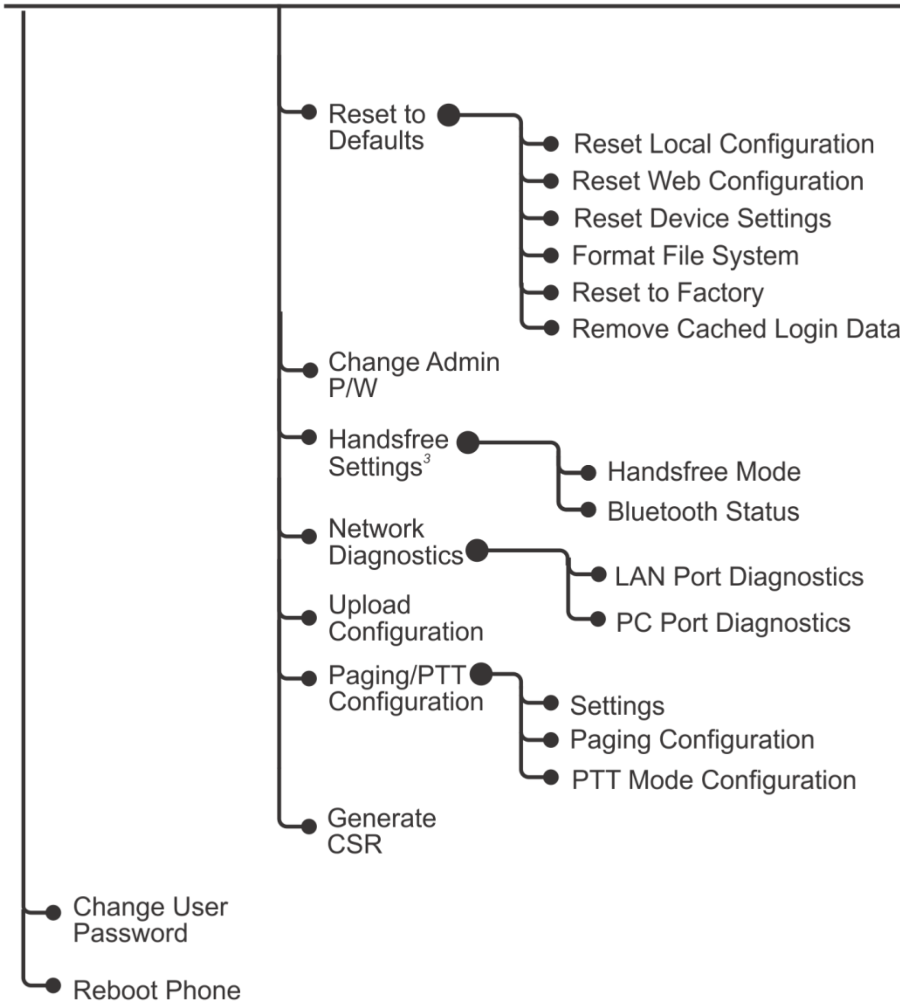
[2] *If enabled.*
[3] *Platform dependent.*

Settings > Advanced[5]



[3] Platform dependent.
[4] If applicable.
[5] Requires administrator password.

Settings > Advanced[5] (Continued)



[3] *Platform dependent.*
[5] *Requires administrator password.*

Directories[6]

- Contact Directory
- Corporate[2] Directory
- Call Lists
  - Missed Calls
  - Received Calls
  - Placed Calls
  - Clear

Messages[6]

Applications[6]

[2] *If enabled.*
[6] *Organization dependent.*

Third-Party Software

This section provides copyright statements for third-party software products that have been incorporated into the Polycom® UC Software 5.0 application distribution. Use the table Third-Party Software to locate license information.

Third-Party Software

Product	*License Location*
c-ares	c-ares
dhcp	dhcp 4.0.0-14
droidfonts	droidfonts
Dropbear	Dropbear
eXpat	eXpat
freetype	freetype
gloox	gloox
ILG JPEG	IJG JPEG
libcurl	libcurl
libMng	libMng
liboil	liboil
libpcap	libpcap
libPng V2	libpng V2
libPng	libPng
libSRTP	libSRTP
libssh2	libssh2
ncurses	ncurses
OpenLDAP	OpenLDAP
OpenSSL	OpenSSL

Product	*License Location*
pmap	pmap-29092002
winPcap	WinPcap
wpa_supplicant	wpa_supplicant
zlib	zlib

Some Polycom products may contain open source software, listed in the table Open-Source Software that is licensed under the terms and conditions of the Free Software Foundation's GPL or LGPL licenses. See the *Polycom Voice OFFER of Source for GPL and LGPL Software*.

Open-Source Software

Product	*Product*	*Product*
alsa-lib	gst-openmax	mtd-utils
alsa-utils	gst-plugins-bad	procps
alsasink	gst-plugins-base	tsattach 1.0
BlueZ	gst-plugins-good	tslib
BusyBox	gst-plugins-ugly	uboot
fbset	gstreamer	udev
ffmpeg	libsoup	Webkit
ffmpegdec	libomxil-bellagio	wireless-tools
freetype	libstdc++	wrsv-ltt
glib2	Linux kernel	x-loader
glibc	module-init-tools	

c-ares

dhcp 4.0.0-14

Internet Systems Consortium, Inc.

950 Charter Street

Redwood City, CA 94063

<info@isc.org>

http://www.isc.org/

droidfonts

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and

2. You must cause any modified files to carry prominent notices stating that You changed the files; and

3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing,

shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Dropbear

The majority of code is written by Matt Johnston, under the license below. Portions of the client-mode work are (c) 2004 Mihnea Stoenescu, under the same license:

Copyright (c) 2002-2006 Matt Johnston

Portions copyright (c) 2004 Mihnea Stoenescu

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

LibTomCrypt and LibTomMath are written by Tom St Denis, and are Public Domain.

=====

sshpty.c is taken from OpenSSH 3.5p1,

Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi>, Espoo, Finland

All rights reserved

"As far as I am concerned, the code I have written for this software can be used freely for any purpose. Any derived versions of this software must be clearly marked as such, and if the derived work is incompatible with the protocol description in the RFC file, it must be called by a name other than "ssh" or "Secure Shell". "

=====

loginrec.c

loginrec.h

atomicio.h

atomicio.c

and strlcat() (included in util.c) are from OpenSSH 3.6.1p2, and are licensed under the 2 point BSD license.

loginrec is written primarily by Andre Lucas, atomicio.c by Theo de Raadt.

strlcat() is (c) Todd C. Miller

=====

Import code in keyimport.c is modified from PuTTY's import.c, licensed as follows:

PuTTY is copyright 1997-2003 Simon Tatham.

Portions copyright Robert de Bath, Joris van Rantwijk, Delian Delchev, Andreas Schultz, Jeroen Massar, Wez Furlong, Nicolas Barry, Justin Bradford, and CORE SDI S.A.

eXpat

freetype

Portions of this software are copyright © <year> The FreeType Project (www.freetype.org). All rights reserved.

gloox

Portions of this SOFTWARE PRODUCT are © 2006 by Jakob Schroeter <js@camaya.net>. All rights reserved.

IJG JPEG

Independent JPEG Group's free JPEG software

This package contains C software to implement JPEG image encoding, decoding, and transcoding. JPEG is a standardized compression method for full-color and gray-scale images.

The distributed programs provide conversion between JPEG "JFIF" format and image files in PBMPLUS PPM/PGM, GIF, BMP, and Targa file formats. The core compression and decompression library can easily be reused in other programs, such as image viewers. The package is highly portable C code; we have tested it on many machines ranging from PCs to Crays.

We are releasing this software for both noncommercial and commercial use. Companies are welcome to use it as the basis for JPEG-related products. We do not ask a royalty, although we do ask for an acknowledgement in product literature (see the README file in the distribution for details). We hope to make this software industrial-quality --- although, as with anything that's free, we offer no warranty and accept no liability.

For more information, contact jpeg-info@jpegclub.org.

Contents of this directory

jpegsrc.vN.tar.gz contains source code, documentation, and test files for release N in Unix format.

jpegsrN.zip contains source code, documentation, and test files for release N in Windows format.

jpegaltui.vN.tar.gz contains source code for an alternate user interface for cjpeg/djpeg in Unix format.

jpegaltuiN.zip contains source code for an alternate user interface for cjpeg/djpeg in Windows format.

wallace.ps.gz is a PostScript file of Greg Wallace's introductory article about JPEG. This is an update of the article that appeared in the April 1991 Communications of the ACM.

jpeg.documents.gz tells where to obtain the JPEG standard and documents about JPEG-related file formats.

jfif.ps.gz is a PostScript file of the JFIF (JPEG File Interchange Format) format specification.

jfif.txt.gz is a plain text transcription of the JFIF specification; it's missing a figure, so use the PostScript version if you can.

TIFFTechNote2.txt.gz is a draft of the proposed revisions to TIFF 6.0's JPEG support.

pm.errata.gz is the errata list for the first printing of the textbook "JPEG Still Image Data Compression Standard" by Pennebaker and Mitchell.

jdosaobj.zip contains pre-assembled object files for JMEMDOSA.ASM.

If you want to compile the IJG code for MS-DOS, but don't have an assembler, these files may be helpful.

libcurl

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2008, Daniel Stenberg, <daniel@haxx.se>.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

libMng

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Gerard Juyn

The MNG Library is supplied "AS IS". The Contributing Authors disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the MNG Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors specifically permit, without fee, and encourage the use of this source code as a component to supporting the MNG and JNG file format in commercial products. If you use this source code in a product, acknowledgment would be highly appreciated.

liboil

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The source code in the liboil/motovec directory is subject to the following license:

You are hereby granted a copyright license to use, modify, and distribute the SOFTWARE so long as this entire notice is retained without alteration in any modified and/or redistributed versions, and that such modified versions are clearly identified as such.

No licenses are granted by implication, estoppel or otherwise under any patents or trademarks of Motorola, Inc.

The SOFTWARE is provided on an "AS IS" basis and without warranty.

To the maximum extent permitted by applicable law, MOTOROLA DISCLAIMS ALL WARRANTIES WHETHER EXPRESS OR IMPLIED, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY AGAINST INFRINGEMENT WITH REGARD TO THE SOFTWARE (INCLUDING ANY MODIFIED VERSIONS THEREOF) AND ANY ACCOMPANYING WRITTEN MATERIALS.

To the maximum extent permitted by applicable law, IN NO EVENT SHALL MOTOROLA BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS) ARISING OF THE USE OR INABILITY TO USE THE SOFTWARE. Motorola assumes no responsibility for the maintenance and support of the SOFTWARE.

libpcap

INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions Copyright (c) 2000 Torsten Landschoff <torsten@debian.org>

Sebastian Krahmer <krahmer@cs.uni-potsdam.de>

License: BSD

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Modifications: Added PACKET_MMAP support

Paolo Abeni <paolo.abeni@email.it>

based on previous works of:

Simon Patarin <patarin@cs.unibo.it>

Phil Wood <cpw@lanl.gov>

Monitor-mode support for mac80211 includes code taken from the iw command; the copyright notice for that code is

Copyright (c) 2007, 2008 Johannes Berg

Copyright (c) 2007 Andy Lutomirski

Copyright (c) 2007 Mike Kershaw

Copyright (c) 2008 GÃ¡bor Stefanik

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions Copyright 1989 by Carnegie Mellon.

Permission to use, copy, modify, and distribute this program for any purpose and without fee is hereby granted, provided that this copyright and permission notice appear on all copies and supporting documentation, the name of Carnegie Mellon not be used in advertising or publicity pertaining to distribution of the program without specific prior permission, and notice be given in supporting documentation that copying and distribution is by permission of Carnegie Mellon and Stanford University. Carnegie Mellon makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

libpng V2

libPng

1. The origin of this source code must not be misrepresented.

2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.

3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

Libpng is OSI Certified Open Source Software. OSI Certified Open Source is a certification mark of the Open Source Initiative.

Glenn Randers-Pehrson

glennrp at users.sourceforge.net

September 10, 2009

libSRTP

Copyright (c) 2001-2005 Cisco Systems, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the Cisco Systems, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

libssh2

Copyright (c) 2004-2007 Sara Golemon <sarag@libssh2.org>

Copyright (C) 2006-2007 The Written Word, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the copyright holder nor the names of any other contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,

STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

ncurses

Copyright (c) 1998-2004, 2006 Free Software Foundation, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, distribute with modifications, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished - to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

OpenLDAP

The OpenLDAP Public License

Version 2.8, 17 August 2003

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

1: Redistributions in source form must retain copyright statements and notices,

2: Redistributions in binary form must reproduce applicable copyright statements and notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution, and

3: Redistributions must contain a verbatim copy of this document.

The OpenLDAP Foundation may revise this license from time to time.

Each revision is distinguished by a version number. You may use this Software under terms of this license revision or under the terms of any subsequent revision of the license.

THIS SOFTWARE IS PROVIDED BY THE OPENLDAP FOUNDATION AND ITS CONTRIBUTORS ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OPENLDAP FOUNDATION, ITS CONTRIBUTORS, OR THE AUTHOR(S) OR OWNER(S) OF THE SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The names of the authors and copyright holders must not be used in advertising or otherwise to promote the sale, use or other dealing in this Software without specific, written prior permission. Title to copyright in this Software shall at all times remain with copyright holders.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Copyright 1999-2003 The OpenLDAP Foundation, Redwood City, California, USA. All Rights Reserved. Permission to copy and distribute verbatim copies of this document is granted.

OpenSSL

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open

The licence and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

pmap-29092002

Copyright (c) 2002 Andrew Isaacson <adi@hexapodia.org>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

WinPcap

Copyright (c) 1999 - 2005 NetGroup, Politecnico di Torino (Italy).

Copyright (c) 2005 - 2010 CACE Technologies, Davis (California).

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the Politecnico di Torino, CACE Technologies nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes software developed by the University of California, Lawrence Berkeley Laboratory and its contributors.

This product includes software developed by the Kungliga Tekniska Hšgskolan and its contributors.

This product includes software developed by Yen Yen Lim and North Dakota State University.

Portions Copyright (c) 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes software developed by the University of California, Berkeley and its contributors."

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions Copyright (c) 1983 Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by the University of California, Berkeley. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Portions Copyright (c) 1995, 1996, 1997 Kungliga Tekniska Hšgskolan (Royal Institute of Technology, Stockholm, Sweden). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes software developed by the Kungliga Tekniska Hšgskolan and its contributors."

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions Copyright (c) 1997 Yen Yen Lim and North Dakota State University. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes software developed by Yen Yen Lim and North Dakota State University"

4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

wpa_supplicant

zlib

version 1.2.3, July 18th, 2005

Jean-loup Gailly   Mark Adler

jloup@gzip.org   madler@alumni.caltech.edu

LICENSE AGREEMENT FOR DEVELOPMENT PURPOSES

This License Agreement for Development Purposes (the "Agreement") is a legal agreement between you and Polycom, Inc., a Delaware corporation ("Polycom").

The software you are about to download (the "Software") comprises sample code that may be useful in the development of applications designed to operate on or in conjunction with Polycom Products.

Polycom is willing to license the Software to you only upon the condition that you accept all of the terms contained in this agreement. Select the "Accept" button at the bottom of the page to confirm your acceptance. If you are not willing to be bound by these terms, select the "Do Not Accept" button and the downloading process will not continue.

PLEASE NOTE:

* POLYCOM OFFERS NO SUPPORT FOR THIS SOFTWARE, AND THE SOFTWARE IS BEING LICENSED WITHOUT DOCUMENTATION, WITHOUT WARRANTY, "AS-IS," AND "WITH ALL FAULTS."

* THE SOFTWARE HAS NOT BEEN TESTED BY POLYCOM AND SHOULD NOT BE LOADED ON PRODUCTION SYSTEMS.

1. GRANT OF LICENSE.

1.1. License. Subject to the terms of this Agreement, Polycom grants to you a nonexclusive, nontransferable license to copy, install, use, and modify the Software, including the Software in source code format, and to produce your own commercial or other purposes derivative works thereof. Except as provided below, this License Agreement does not grant you any rights to patents, copyrights, trade secrets, trademarks, or any other rights related to the Software.

2. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS.

2.1. Copyright. All title and copyrights in and to the Software and any copies of the Software are owned by Polycom or its suppliers. The Software is protected by copyright laws and international treaty provisions. Title, ownership rights, and intellectual property rights in the Software shall remain in Polycom or its suppliers.

2.2. Ownership of Derivative Works. As between you and Polycom, you will own copyright and other intellectual property rights in derivative works of the Software that you develop.

2.3. Reservation. Polycom reserves all rights in the Software not expressly granted to you in this Agreement.

3. SUPPORT SERVICES.

3.1. No Support Services. Polycom provides no support services for the Software.

4. TERMINATION.

4.1. Termination. Without prejudice to any other rights, Polycom may terminate this Agreement if you fail to comply with any of the terms and conditions of this Agreement. In such event, you must destroy all copies of the Software and all of its component parts. You may terminate this Agreement at any time by destroying the Software and all of its component parts.

5. NO WARRANTY.

THE SOFTWARE IS LICENSED WITHOUT WARRANTY, "AS IS," AND "WITH ALL FAULTS." ALL WARRANTIES, TERMS OR CONDITIONS, EXPRESS OR IMPLIED, EITHER IN FACT OR BY OPERATION OF LAW, STATUTORY OR OTHERWISE, INCLUDING WARRANTIES, TERMS OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, SATISFACTORY QUALITY, CORRESPONDENCE WITH DESCRIPTION, AND NON-INFRINGEMENT, ARE EXPRESSLY DISCLAIMED. POLYCOM NEITHER ASSUMES NOR AUTHORIZES ANY OTHER PERSON TO ASSUME FOR IT ANY OTHER LIABILITY IN CONNECTION WITH THE SALE, INSTALLATION, MAINTENANCE, OR USE OF THIS SOFTWARE.

6. LIMITATION OF LIABILITY.

6.1. Limitations. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL POLYCOM OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING. WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE, EVEN IF POLYCOM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY CASE, POLYCOM'S ENTIRE LIABILITY SHALL BE LIMITED TO THE GREATER OF THE AMOUNT ACTUALLY PAID BY YOU FOR THE SOFTWARE OR U.S. $5.00.

7. DISCLAIMER.

7.1. Disclaimer. Some countries, states, or provinces do not allow the exclusion or limitation of implied warranties or the limitation of incidental or consequential damages for certain products supplied to consumers or the limitation of liability for personal injury, so the above limitations and exclusions may be limited in their application to you.

8. EXPORT CONTROLS.

8.1. Export Controls. The Software may not be downloaded or otherwise exported or re-exported (i) into (or to a national or resident of) Cuba, Iraq, Libya, North Korea, Yugoslavia, Iran, Syria, Republic of Serbia, or any other country to which the U.S. has embargoed goods; or (ii) to anyone on the U.S Treasury Department's List of Specially Designated Nationals or the U.S. Commerce Department's Table of Denial Orders. By downloading or using this Software, you are agreeing to the foregoing and you are representing and warranting that you are not located in, under the control of, or a national or resident of any such country or on any such list. If you obtained this Software outside of the United States, you are also agreeing that you will not export or re-export it in violation of the laws of the country in which it was obtained.

9. MISCELLANEOUS.

9.1. Governing Law. This Agreement shall be governed by the laws of the State of California as such laws are applied to agreements entered into and to be performed entirely within California between California residents, and by the laws of the United States. The United Nations Convention on Contracts for the International Sale of Goods (1980) is hereby excluded in its entirety from application to this Agreement.

9.2. Venue for Resolving Disputes. Any disputes relating to this Agreement will be resolved only in the state or federal courts located in Santa Clara County, California. Each of the parties agrees to the exercise over them of the personal jurisdiction of such courts for such purpose.

9.3. U.S. Government Restricted Rights. The Software and documentation are provided with Restricted Rights. The Software programs and documentation are deemed to be "commercial computer software" and "commercial computer software documentation," respectively, pursuant to DFAR Section 227.7202 and FAR 12.212(b), as applicable. Any use, modification, reproduction, release, performance, display, or disclosure of the Software programs and/or documentation by the U S. Government or any of its agencies shall be governed solely by the terms of this Agreement and shall be prohibited except to the extent expressly permitted by the terms of this Agreement. Any technical data provided that is not covered by the above provisions is deemed to be "technical data commercial items" pursuant to DFAR Section 227.7015(a). Any use, modification, reproduction, release, performance, display, or disclosure of such technical data shall be governed by the terms of DFAR Section 227.7015(b).

9.4. Relationship Between the Parties. The relationship between you and Polycom is that of licensee/licensor. Neither party will represent that it has any authority to assume or create any obligation, express or implied, on behalf of the other party, nor to represent the other party as agent, employee, franchisee, or in any other capacity. Nothing in this agreement shall be construed to limit either party's right to independently develop or distribute software that is functionally similar to the other party's products, so long as proprietary information of the other party is not included in such software.

9.5. Entire Agreement. This Agreement represents the complete agreement concerning this license and may be amended only by a writing executed by both parties. If any provision of this Agreement is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

www.polycom.com

Corporate Headquarters: Polycom, Inc. 6001 America Center Drive San Jose, CA 95002 USA