Polycom© UC Software 5.5.0

Contents

Session Initiation Protocol (SIP)

Monitoring, Diagnostics, and Troubleshooting

Tables

Figures

Get Started

The Polycom® Unified Communications (UC) Software Administrator Guide provides general guidance on installing, provisioning, and managing Polycom phones. This guide helps you:

- Set up a large-scale provisioning environment
- Set up a VoIP network and provisioning server
- Configure phone features and user settings
- Troubleshoot, update, and maintain phones and UC Software

This Polycom UC Software Administrator Guide applies to the following Polycom devices except where noted:

- Polycom VVX business media phones, including the VVX 101, 201, 300 series (300/301/310/311), 400 series (400/401/410/411), 500 series (500/501) 600 series (600/601), and 1500
- Polycom VVX D60 Wireless Handset and Base Station
- Polycom VVX Expansion Modules
- Polycom SoundStructure® VoIP Interface

Polycom phones with Microsoft servers

To deploy Polycom devices with Microsoft Lync Server or Skype for Business, see *Polycom UC Software with Microsoft Lync Server and Skype for business - Deployment Guide* on Polycom Voice Support.

Audience and Purpose of This Guide

The primary audience for this guide is the person administering the session initiation protocol (SIP) server, provisioning servers, VoIP network, and Polycom UC Software that enable you to configure and manage phone features. This guide is not intended for end users. This guide provides information primarily for mid-level administrators with experience in networking who understand the basics of open SIP networks and VoIP endpoint environments. This guide indicates where information might be useful for novice administrators, and provides tips for advanced administrators where applicable.

Phone Deployment Scenarios

Because phone deployments vary, and administrators typically set up and maintain large-scale device deployments, Polycom cannot recommend a specific deployment scenario. For large-scale deployments, Polycom recommends setting up a provisioning server on the local area network (LAN) or on the Internet. For this reason, this administrator guide focuses on large-scale UC Software VoIP environments set up on

a central SIP and provisioning server. Administrators typically use the administrator guide in three large-scale device deployment scenarios:

● **Enterprise deployment**. An administrator sets up and maintains a deployment for a single organization and all users are in one physical location.

● **Multisite enterprise**. An administrator sets up and maintains a deployment for an organization and users are spread out over several locations varying in size.

● **Service Provider Deployment**. Service providers provide devices and service to a number of organizations and users spread out over several locations each varying in size.

Requirements

This section lists the general knowledge, skill, and technological requirements needed to deploy and provision Polycom devices and configure features.

General Knowledge Requirements

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

● Computer networking and driver administration for your operating system

● SIP networks

● VoIP environments and technologies

● An XML editor

Polycom-Specific Skills

You require the following Polycom-specific skills to successfully deploy and configure Polycom devices:

● Polycom provisioning methods

● Polycom UC Software and XML configuration files

● Configuration parameters and values for end-user device features

● Troubleshooting your Polycom devices

● Maintaining and updating devices and software

Technological Requirements

You require the following to operate Polycom phones as SIP endpoints in large-scale deployments:

● 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 Polycom Desktop Phone Compatibility. If you are using the Polycom RealPresence Trio Solution, see Polycom RealPresence Trio and SoundStation IP Platform Compatibility. At minimum, your call server requires:

  ➢ A call server address that registers voice endpoints with the SIP server

➢ SIP authentication user name and password the phone uses to respond to any SIP authentication challenges from the SIP server.

● An XML editor—such as XML Notepad—to create and edit configuration files

Get Help

For more information about installing, configuring, and administering Polycom products, refer to Documents and Downloads at Polycom Support.

To access Polycom UC Software releases and documentation, see Polycom Voice Support.

To access the user guide for Polycom voice products, refer to the product support page for your phone at Polycom Voice Support.

To find help or technical support for your phones, you can search for Polycom documentation at the Polycom Unified Communications (UC) Software Resource Center.

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

The Polycom Community

The Polycom Community gives you access to the latest developer and support information and enables you to participate in discussion forums to share ideas and solve problems with your colleagues. To register with the Polycom Community, create a Polycom online account. When logged in, you can access Polycom support personnel and participate in developer and support forums to find the latest information on hardware, software, and partner solutions topics.

Polycom | Support

For support or service, please contact your Polycom reseller or visit support.polycom.com for software downloads, product documents, product licenses, troubleshooting tips, service requests, and more.

We are constantly working to improve the quality of our documentation, and we would appreciate your feedback. Please send email to VoiceDocumentationFeedback@polycom.com.
Polycom recommends that you record the phone model numbers, software versions (for both the Updater and UC Software), and partner platform for future reference.

Phone models:

Updater version:

UC Software version:

Partner Platform:

Provision and Configure Phones with Polycom UC Software

This section provides an overview of how to deploy Polycom phones using Polycom UC Software and includes the following major topics:

- Polycom UC Components
- Set Up a Network for Polycom UC Software
- Polycom Provisioning Methods
- Set Up a Provisioning Server
- Deploy Devices from a Provisioning Server
- Master Configuration File Fields
- Configure with the Master Configuration File

Polycom UC Components

This section is intended for administrators not familiar with Polycom UC Software. This section provides general information about the following UC Software components:

- The Updater
- The Polycom UC Software File Image
- XML Resource Files, Configuration Templates, and the XML Schema File

The Updater

The Updater is a small application that resides in the flash memory on the phone. Polycom phones come installed with the Updater.

When you start/boot/reboot the phone, the Updater automatically performs the following tasks:

1 The setup menu displays so you can set various network and provisioning options.

    The Updater requests IP settings and accesses the provisioning server (also called the boot server) to look for changes to the Updater software. If updates are found, they are downloaded and saved to flash memory, which overwrites itself after verifying the integrity of the download.

2 If new updates are downloaded, the Updater formats the file system, removes any application software and configuration files that were present.

3 The Updater downloads the master configuration file.

    The Updater and the application use this file to acquire a list of other files that the phone needs.

4   The Updater examines the master configuration file for the name of the application file, and then looks for this file on the provisioning server.

If the copy on the provisioning server is different from the one stored in device settings, or there is no file stored in flash memory, the application file is downloaded.

5   The Updater extracts the Polycom UC Software from flash memory.

6   The Updater installs the application into RAM, and then uploads an event log file from the boot cycle.

7   The Updater completes the cycle, and the Polycom UC Software begins running the phone's operations.

The Polycom UC Software File Image

Polycom UC Software is a binary file image and contains a digital signature that prevents tampering or the loading of rogue software images. Each release of software is a new image file. Both the Updater and Polycom UC Software run on all Polycom device models.

Polycom UC Software manages the protocol stack, the digital signal processor (DSP), the user interface, the network interaction, and implements 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 and H.323 signaling for video telephony.

● Industry standard security techniques for ensuring that all provisioning, signaling, and media transactions are robustly authenticated and encrypted.

● Advanced audio signal processing for handset, headset, and speakerphone communications using a wide range of audio codecs.

● Flexible provisioning methods to support single phone, small business, and large multi-site enterprise deployments.

XML Resource Files, Configuration Templates, and the XML Schema File

Polycom UC Software includes a number of resource files, template configuration files, and an XML schema file that provides examples of parameter types and permitted value types. The resource and configuration files contains parameters you can use to configure features and apply settings to phones. Configuration files are for use with the centralized provisioning method as explained in Centralized Provisioning.

Resource Files

The UC Software download contains optional resource configuration files you can apply to the phones. In addition, you can allow phone-specific override files containing user settings to be uploaded to the central server. Resource and override files include:

● Language dictionaries for the phone menu and Web Configuration Utility

● Configuration override files that store settings made from the phone menu and Web Configuration Utility. To allow override files to be uploaded to the central server, refer to Setting Server Permissions for Override Files.

● Ringtones

● Log files

- A template contact directory 000000000000-directory~.xml
- A licensing directory

Configuration Templates

The following table lists the template directories and files included in the UC Software download.

Note that `techsupport.cfg` is available from Polycom Customer Support for troubleshooting and debugging.

Configuration File Templates

Name	Description	Deployment Scenarios
Directories		
PartnerConfig	Contains configuration file specific to the following third-party servers: • Alcatel-Lucent • BroadSoft • GENBAND • Microsoft • Sylantro	For use with third-party servers.
Config		
applications.cfg	For applications, browser, microbrowser, XMP-API	Typical Hosted Service Provider Typical IP-PBX
device.cfg	Network Configuration parameters	Troubleshooting Administrative settings
features.cfg	Features including corporate directory, USB recording, presence, ACD	Typical Hosted Service Provider Typical IP-PBX
firewall-nat.cfg	Firewall parameters	
H323.cfg	H.323 video use	Typical Hosted Service Provider using VVX 500/501, 600/601, and 1500 for video calls
lync.cfg	Microsoft Skype for Business parameters	Typical Microsoft Skype for Business environment
polycomConfig.xsd*	See XML Resource Files, Configuration Templates, and the XML Schema File	
pstn.cfg		
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

Configuration File Templates

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.	Use for troubleshooting and debugging only
video.cfg	VVX 500/501, 600/601, and 1500 video	Typical Hosted Service Provider if using VVX 500/501, 600/601, and 1500 for video calls
video-integration.cfg		

Using Correct Parameter XML Schema, Value Ranges, and Special Characters

The configuration parameters available in the UC Software templates use a variety of values, including Boolean, integer, enumerated types, and arrays (a table of values). UC Software includes an XML schema file—`polycomConfig.xsd`—that provides information about parameter type, permitted values, default values, and valid enumerated type values. View this template file with an XML editor.

Polycom configuration parameters support the following value types:

- Boolean
- Enumerated
- Integer
- String

The following rules apply to UC Software parameter values:

- Boolean values are not case sensitive.
- UC Software interprets `Null` as empty.
- The values `0,` `false,` and `off` are supported and inter-changeable.
- The values `1,` `true,` and `on` are supported and inter-changeable. This administrator 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 value is greater than the allowable range, the maximum allowable value is used.
- If the value is less than the allowable range, the minimum allowable value is used.
- If you insert invalid parameter values into the configuration file, the value is ignored and the default value is used. Examples of invalid parameter values include enumerated values that do not match values defined in the UC Software, numeric parameters set to non-numeric values, string parameters whose value is too long or short, null strings in numeric fields. Invalid values are logged in the phone's log files.

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;`

Set Up a Network for Polycom UC Software

A typical large-scale VoIP deployment requires administrators to complete each of the following tasks. Note that deployment scenarios vary and Polycom cannot recommend a specific environment.

To set up a centralized provisioning environment:

1 Create user accounts on the SIP call server.

2 (Optional) Set up a provisioning server. In some cases a provisioning server is built into the SIP call server and if not, administrators must set up their own provisioning server. For details, refer to Set Up a Provisioning Server.

 Polycom strongly recommends setting up a provisioning server for large-scale VoIP device deployments. 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.

3 (Optional) Configure security options on your network.

 ➢ 802.1X

 ➢ Virtual local area networks (VLANs)

 ➢ File transfers using HTTPS

 ➢ SIP signaling over Transport Layer Security (TLS)

 ➢ Set permissions for configuration and override files. Refer to Setting Server Permissions for Override Files.

4 Set up Dynamic Host Configuration Protocol (DHCP).

5 Set up Domain Name System (DNS). Polycom supports the following DNS records types:

 ➢ DNS A record

 ➢ Service (SRV) record for redundancy

 ➢ Name Authority Pointer (NAPTR)

6 Connect the phones to the network.

7 Deploy phones from the provisioning server as shown in Deploy Devices from a Provisioning Server.

Polycom Provisioning Methods

Polycom provides several methods to provision phones. The method you use depends on the number of phones and how you want to apply features and settings. Methods available can vary by device model.

You can use multiple methods concurrently to provision and configure features but there is a priority among the methods when you use multiple methods concurrently—settings you make using a higher priority configuration method override settings made using a lower priority method. When using multiple configuration methods, a setting you make using a lower-priority method does not apply to or override a duplicate setting made using a higher-priority method. The provisioning and configuration methods in order of priority are as follows:

- Quick Setup
- Phone menu
- Web Configuration Utility
- Centralized provisioning

Note that features and settings vary by method, by device, and by UC Software release. For this reason, Polycom recommends limiting the methods you use concurrently to avoid confusion about where a phone is receiving settings.

Quick Setup of Polycom Phones

By default, Quick Setup is enabled on phones and the QSetup soft key displays on the phone interface when the phone is booting. This key allows users to access the provisioning server and configure the phone for provisioning.

After the user completes initial configuration, you can show or hide the QSetup soft key using the parameter in the following table.

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

Configure Quick Setup

This section lists parameters that configure the Quick Setup feature.

Quick Setup Soft Key Parameter

Parameter Template	Permitted Values
`prov.quickSetup.enabled` site.cfg	1 (default) - The quick setup feature and soft key is enabled. 0 - The quick setup feature and soft key is disabled.

Provision Using the Phone Menu

You can use the phone menu system to provision a single phone and to configure features on one phone. If you are provisioning more than about 10 to 20 phones, Polycom recommends using centralized provisioning as your primary provisioning method.

You can use the menu system as the sole configuration method or in conjunction with other methods. Menu systems and interface settings vary by device and by UC Software release. Settings you make from the phone menu override settings you make using the Web Configuration Utility and central provisioning server. However, the phone menu does not contain all of the settings available with centralized provisioning.

The phone menu system makes settings available to users and administrators; settings available to administrators only can be accessed on the Advanced menu and require an administrator password. For information on setting passwords, see Set Local User and Administrator Passwords.

If you need to reset all settings made from the device interface to default, refer to the section Reset the Phone to Defaults with an MKC.

Provision Using the Web Configuration Utility

The Web Configuration Utility is a web-based interface that is especially useful for remote provisioning and configuration. You can use the Web Configuration Utility to provision a single phone and to configure features on one device. If you are provisioning more than about 10 to 20 phones, Polycom recommends using centralized provisioning as your primary provisioning method.

You can use the Web Configuration Utility as the sole configuration method or in conjunction with other methods. Because features and settings can vary by device model and UC Software release, options available in the Web Configuration Utility can vary. In addition, the Web Configuration Utility does not contain all of the settings available with centralized provisioning. Settings you make from the Web Configuration Utility override settings you make on the central provisioning server and do not override settings you configure from the phone menu system. If you want to remove settings applied from the Web Configuration Utility, click the Reset to Default button on any page in the Web Configuration Utility.

For more detailed help using the Web Configuration Utility, see the *Polycom Web Configuration Utility User Guide* on Polycom UC Software Support Center.

The Web Configuration Utility makes settings available to users and administrators; settings available to administrators only can be accessed on the Advanced menu and require an administrator password. For information on setting passwords, see Set Local User and Administrator Passwords.

> **Note: 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 information, see *Feature Profile 67993: Using the Software Upgrade Tool in the Web Configuration Utility* on Polycom Profiled UC Software Features.

Centralized Provisioning

This section provides important points about using Polycom UC Software in large-scale deployments.

● Centralized provisioning enables you to provision phones from a provisioning server that you set up, and maintain a set of configuration files for all phones on a central provisioning server. The centralized provisioning method is recommended for phone deployment of about 20 or more phones. After phones are provisioned with UC Software, you can configure features and settings for all phones with the UC Software configuration files that you store and modify on your provisioning server. For information about configuring features and settings, refer to Configure with the Master Configuration File

● Most configuration parameters are located in only one template file; however, some are included in two or more files. The template configuration files are flexible: you can rearrange the parameters within the template, move parameters to new files, or create your own configuration files from

parameters you want. This flexibility is especially useful when you want to apply specific settings to a group of phones. You can create and name as many configuration files as you want and your configuration files can contain any combination of parameters. For a list of all template files included in the UC Software, refer to XML Resource Files, Configuration Templates, and the XML Schema File.

● To apply settings to phones, you must write the name of configuration files to the CONFIG_FILES field of the master configuration file. The files you enter to the CONFIG_FILES field write are read from left to right. Duplicate settings are applied from the configuration file in the order you list them. For details about the master configuration file fields, refer to Master Configuration File Fields.

● Polycom phones boot up without the use of configuration files, and you can specify a SIP server address and a registration address (the equivalent of a phone number) in a configuration file before or after the phone boots up. If a phone cannot locate a provisioning server upon boot up, and has not been configured with settings from any other source, the phone operates with internally stored default values. If the phone has been previously configured with settings from a provisioning server and cannot locate the server when booting up, the phone operates with those previous settings.

● If settings you make from the central server are not working, check first for priority settings applied from the phone menu system or Web Configuration Utility, and second for duplicate settings in your configuration files. For information about how override settings are applied, refer to Polycom Provisioning Methods.

Set Up a Provisioning Server

After you set up a VoIP network and create accounts on the SIP call server, shown in Set Up a Network for Polycom UC Software, set up a centralized provisioning server to provision the phones and configure settings. Polycom phones support the FTP, TFTP, HTTP, and HTTPS protocols, and use FTP by default. The example shown in this section uses FTP and a personal computer (PC) as the provisioning server.

> **Note: Use RFC-compliant servers**
> Polycom recommends that you use RFC-compliant servers.

Before you begin, install and set up tools on your PC, and gather some information:

● If using Power over Ethernet (PoE) with the phone, you need a PoE switch and network cable.

● Install an XML editor, such as XML Notepad 2007, on your computer.

● Install an FTP server application on your computer. FileZilla and *wftpd* are free FTP applications for windows and *vsftpd* is typically available with all standard Linux distributions.

● Take note of the following:

  ➢ **SIP Server address**   This is the hostname or IP address of the call server that handles VoIP services on your network.

  ➢ **SIP account information**   This may include SIP credentials such as a user name and password, and the phone's registration address. Although a user name and password are not required to get the phone working, Polycom strongly recommends using them for security reasons.

  ➢ **MAC address**   Each phone has a unique 12-digit serial number just above the phone's bar code on a label on the back of the phone. Collect the MAC address for each phone in your deployment.

  ➢ **Your computer's IP address**   To use your computer as the provisioning boot server, you need your computer's IP address. Jot this number down as you need it at the end of the provisioning process.

To set up the provisioning server:

1 Provide power to the phone using a PoE switch, if available, or, if no PoE switch is available, using an external power adapter and a network cable to connect the phone to your network.

2 Install and set up an FTP application. FileZilla and *wftpd* are free FTP applications for windows and *vsftpd* is typically available with all standard Linux distributions.

   You must create a root FTP directory on the provisioning computer with full read and write access to all directories and files. You will be placing configuration files in this root directory.

   In your FTP server application, create a user account for the phone to use and take note of the user name and password as you will need these later in the provisioning process. Launch the FTP application and keep it running at all times so that the phones can communicate with the UC Software.

3 Download the UC software version(s) to your root directory from the Polycom UC Software Support Center. To match a phone model with a correct Polycom UC Software release, refer to the Polycom UC Software Release Matrix for VVX Phones and SoundStructure.

   You can choose the combined UC Software package or the split UC Software package, both in ZIP file format.

   ➢ The combined version contains all files for all phone models.

   ➢ The split software package is smaller, downloads more quickly, and contains **sip.ld** files for each phone model, enabling you to choose provisioning software for your phone model(s) and maintain software versions for each model in the same root directory.

4 To apply security settings to your configuration files, refer to the section Encrypt Configuration Files.

Configure Multiple Servers

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. You can use the table to configure the number of times each server is 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.

Setting Server Permissions for Override Files

By default, phones you provisioning from a central server attempt to upload a number of phone-specific files to the server. If you want to allow the phone to upload these files to the server, you must have read, write, and delete permissions on the server account and provide enable, read, and write access to those files. To organize these files, Polycom recommends creating a separate directory on the server for each file type you want to allow uploads for:

● Log files

● Configuration override files from the local phone interface and Web Configuration Utility. For more information on override files, refer to Override Files.

● A contact directory

● A license directory

Each directory can have different access permissions, for example, you can allow log, contacts, and overrides to have full read and write access, and a license directory to have read-only access. However, where the security policy permits, Polycom recommends allowing these file uploads to the provisioning server to allow greater manageability and can help Polycom provide customer support when diagnosing issues with the phone. All other files that the phone needs to read, such as the application executable and standard configuration files, should be read-only. Ensure that the file permissions you create provide the minimum required access and that the account has no other rights on the server.

Note that as of Polycom UC Software 4.0.0, you can create user-specific configuration files that enable phone users to use their features and settings from any phone in an organization. For instructions, refer to the section Set User Profiles.

Override Files

When settings are modified from the phone user interface or Web Configuration Utility (user or administrator), the phone attempts to upload override files with settings to the central server. When using a central provisioning server as part of your VoIP environment, you have the option to store the override file to the phone, or you can permit the phone to upload the override file to the provisioning server by giving the phone write access to the provisioning server. Allowing the phone access to the provisioning server enables user settings to survive restarts, reboots, and software upgrades administrators apply to all phones from the provisioning server.

You can also use the override files to save user custom preferences and to apply specific configurations to a device or device group. If you permit the phone to upload to the provisioning server, the override file is by default named either <MAC Address>-phone.cfg or <MAC Address>-Web.cfg depending on the whether the change was made from the phone or Web Configuration Utility respectively.

If you reformat the phone's file system, the override file is deleted. If you need to clear phone settings and features applied by override files, refer to Update Configuration from the Phone Menu.

Deploy Devices from a Provisioning Server

After setting up your provisioning server(s), you can deploy devices. This section shows you how to deploy your Polycom devices from the provisioning server using Polycom UC Software.

> **Note: Configuring your phone for local conditions**
> If SNTP settings are not available through DHCP, you need to edit the SNTP GMT offset, and possibly the SNTP server address for the correct local conditions. Changing the default daylight savings parameters might be necessary outside of North America. If the local security policy dictates you might need to disable the local Web (HTTP) server or change its signaling port. To change the default location settings for user interface language and time and date format.

To deploy phones with a provisioning server:

1 Using the list of MAC addresses of each phone you are deploying, create a per-phone **phone<MACaddress>.cfg** file.

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

2 Add the SIP server registration information and user account information to parameters in the per-phone file, for example `reg.1.address`, `reg.1.auth.userId`, `reg.1.auth.password`, `reg.1.label`, `reg.1.type`.

3 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`.

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

a Enter the name of each per-phone and per-site configuration file 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, refer to Master Configuration File Fields and Configure with the Master Configuration File.

For example, add a reference to **phone<*MACaddress*>.cfg** and **sipVVX500.cfg**.

b (Optional) Edit the `LOG_FILE_DIRECTORY` attribute of master configuration file to point to the log file directory.

c (Optional) Edit the `CONTACT_DIRECTORY` attribute of master configuration file to point to the organization's contact directory.

(Optional) Edit the `USER_PROFILES_DIRECTORY` attribute of master configuration file if you intend to enable the user login feature. For more information, see the section Set User Profiles.

d (Optional) Edit the `CALL_LISTS_DIRECTORY` attribute of master configuration file to point to the user call lists.

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

a On the phone's **Home** screen or idle display, select **Settings > Advanced > Admin Settings > Network Configuration > Provisioning Server**. When prompted for the administrative password, enter **456**.

The Provisioning Server entry is highlighted.

b Press 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 Press 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 Press the **Back** soft key twice.

h Scroll down to **Save & Reboot**, and then press the **Select** soft key.

The phone reboots and the UC Software modifies the `APPLICATION APP_FILE_PATH` attribute of the master configuration file so that it references the appropriate sip.ld files.

After this step, the UC Software reads the unmodified `APPLICATION APP_FILE_PATH` attribute. Then, the phone sends a DHCP Discover packet to the DHCP server. You can locate this in the Bootstrap Protocol/option 'Vendor Class Identifier' section of the packet which includes the phone's part number and the BootROM version. For more information, see the section Parse Vendor ID Information.

6 Ensure that the configuration process completed correctly:

a On the phone, press **Settings** (**Menu** if using a VVX 1500) **> Status > Platform > Application > Main** to see the UC Software version and **Status > Platform > Configuration** to see the configuration files downloaded to the phone.

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

The phone uploads two logs files to the LOG_DIRECTORY directory: ***<MACaddress>*app.log** and ***<MACaddress>*boot.log**.

You can now instruct your users to begin making calls.

> **Settings: View the phone's provisioning information**
> To view phone provisioning information, use the multikey shortcut by simultaneously pressing 1-4-7 to display:
> - Phone IP address
> - Phone MAC address
> - VLAN ID
> - Boot server type (FTP, TFTP, HTTP, HTTPS)

Master Configuration File Fields

The centralized provisioning method requires you to use a master configuration file, named **00000000000.cfg** in the UC Software download. Familiarize yourself with the master configuration file fields to use centralized provisioning effectively.

Default fields in the master configuration file
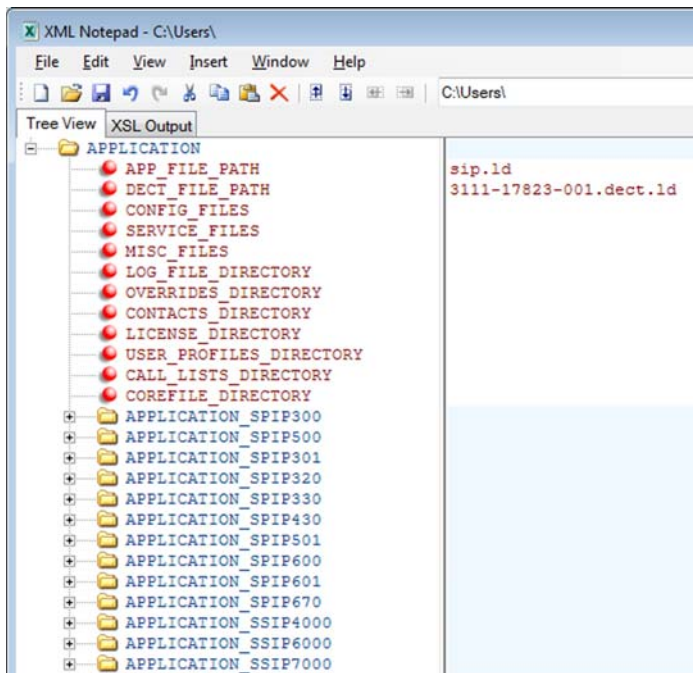
The following describes the XML field attributes in the master configuration file and the APPLICATION directories.

- **APP_FILE_PATH**   The path name of the UC Software application executable. The default value is sip.ld. Note that the phone automatically searches for the sip.ld and <*part number*>.sip.ld. This field can have a maximum length of 255 characters. If you want the phone to search for a sip.ld file in a location other than the default or use a different file name, or both, modify the default. For example, you can specify a URL with its own protocol, user name, and password: http://usr:pwd@server/dir/sip.ld.

- **DECT_FILE_PATH**   The path for the application executable for the Polycom® VVX® D60 Wireless Handset. The default value is 3111-17823-001.dect.ld. When the software for a VVX business media phone with a paired VVX D60 Base Station is updated, the phone also searches for the dect.ld for any updates to the base station software.

    If you want the phone to search for the 3111-17823-001.dect.ld in a location other than the default or use a different file name, or both, modify the default. For example, you can specify a URL with its own protocol, user name, and password: http://usr:pwd@server/dir/3111-17823-001.dect.ld.

- **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`. The files names you enter to the CONFIG_FILES field write are read from left to right. Duplicate settings are applied from the configuration file in the order you list them

- **MISC_FILES**   A comma-separated list of files. Use this to list volatile files that you want phones to download, for example, background images and ringtone .wav files. The phone downloads files you list here when booted, which can decrease access time.

- **LOG_FILE_DIRECTORY**   An alternative directory for log files. You can also specify a URL. This field is blank by default.

- **CONTACTS_DIRECTORY**   An alternative directory for user directory files. You can also specify a URL. This field is blank by default.

- **OVERRIDES_DIRECTORY**   An alternative directory for configuration overrides files. You can also specify a URL. This field is blank by default.

- **LICENSE_DIRECTORY**   An alternative directory for license files. You can also specify a URL. This field is blank by default.

- **USER_PROFILES_DIRECTORY**   An alternative directory for the *<user>*.cfg files.

- **CALL_LISTS_DIRECTORY**   An alternative directory for user call lists. You can also specify a URL. This field is blank by default.

- **COREFILE_DIRECTORY**   An alternative directory for Polycom device core files to use to debug problems. This field 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 your phones.

Alternatively, you can specify the location of a master configuration file you want the phones to use, for example, `http://usr:pwd@server/dir/example1.cfg`. The file name must be at least five characters long and end with **.cfg**. If the phone cannot find and download a location you specify, the phone searches for and uses a per-phone master configuration file and then the default master configuration file.

Configure with the Master Configuration File

The master configuration file maximizes the flexibility you have to customize features and settings for your devices in large deployments. You can use the master configuration file to configure features and apply settings for:

- All phones
- Different groups of phones
- Specific phone models
- A single phone

You can use the default name `000000000000.cfg` or configure features and settings for phone groups by renaming the master configuration file. You can use any of these methods concurrently within the same deployment. There are two ways rename the master configuration file:

- Define a MACaddress.cfg file
- Use a variable substitution

The method you use depends on your deployment scenario and understanding all naming schemes can help you to deploy and manage your phones efficiently.

Find a Phone's MAC Address

Each phone has a unique a-f hexadecimal digit called a MAC address, also known as the serial number (SN). You can use the MAC address to create variables in the name of the master configuration file, or to specify phone-specific configuration files. There are three ways to find a phone's MAC address.

To find a phone's MAC Address:

- » Find the MAC Address by doing one of the following:
  - ➢ Look on the label on the back of the phone
  - ➢ On the phone's menu system, go to **Settings** (**Menu** if using a VVX 1500) **> Status > Platform > Phone > S/N:**
  - ➢ Use a multikey shortcut by simultaneously pressing **1-4-7**

Define a Per-Phone *MACaddress*.cfg File

You can create a *MACaddress.cfg* file for each phone by making a copy and renaming the master configuration file template. Note that you can use only lower-case letters, for example, **0004f200106c.cfg**.

The advantage of using this method is a high degree of control over each phone. If you want to modify or add settings on a per-phone basis, add a new configuration file to the CFG_FILES field of each user-<MACaddress>.cfg phone file or make changes to an existing configuration file.

For large deployments, this naming scheme can require some file management as you need to create and edit at least two unique files for each phone in your deployment, the MACaddress.cfg file and one or more configuration files unique to each phone.

Note: 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, *<MACaddress>*-boot.log, or *<MACaddress>*-license.cfg. These filenames are used by the phone to store override files and logging information.

To create a per-phone MAC address configuration files:

1 Create a copy of the master configuration file template for each phone.

2 Create a *MACaddress.cfg* file for each phone, replacing `000000000000` with the unique MAC address of each phone you are configuring, for example `0004f2123456`.cfg.

   You can find the MAC address of your phone on a label on back of the phone.

3 Create a file for each phone containing information unique to each phone, for example, registration information. You can use the template files in the UC Software download, or you can create your own configuration file using parameters from the UC Software template files. Give your files a name that indicates the file contents or purpose. For example, you might use parameters from the `reg-basic.cfg` template file to create a registration file named `reg-basic_john_doe.cfg`.

4 Enter the name of the file you created in step two in the CONFIG_FILES field of the *MACaddress.cfg* file you created in step one for each phone.

5 Save the master configuration file.

Configure Phones Using a Variable Substitution

This method enables you to configure all phones using a single master configuration file instead of a MACaddress.cfg file for each phone. This method follows from the phone's programmed behavior: the phone looks first for a file containing its own MAC address and if it cannot find that, uses the default *000000000000.cfg* master configuration file.

This method is useful if you need to maintain or modify settings common to all of the phones in your deployment. To apply a common configuration to all phones, you need only create one new configuration file and add it to the CONFIG_FILES field of the 000000000000.cfg master file. If you want to add a new phone to your deployment, you need only create one new file.

For more information on creating phone groups and using variable substitutions, see the section Use a Variable in the Master Configuration File.

To configure using a variable substitution:

1 Create a file for each phone containing information unique to each phone, for example, registration information. The name of this file must contain the phone's unique MAC address, for example, `reg-basic_0004f2000001.cfg`. Each of these phone-specific configuration files must be named identically, varying only in the MAC address of each phone.

2 Enter the name of any one of your phone-specific files to the CONFIG_FILES field of the master configuration file.

3 Modify the file name in the CONFIG_FILES field by replacing the phone-specific MAC address with the variable [PHONE_MAC_ADDRESS] and include the square brackets. You must enter the variable in the same place you entered the phone's MAC address in the phone-specific file.

   For example, modify `reg-basic_0004f2000001.cfg` to `reg-basic_[PHONE_MAC_ADDRESS].cfg`.
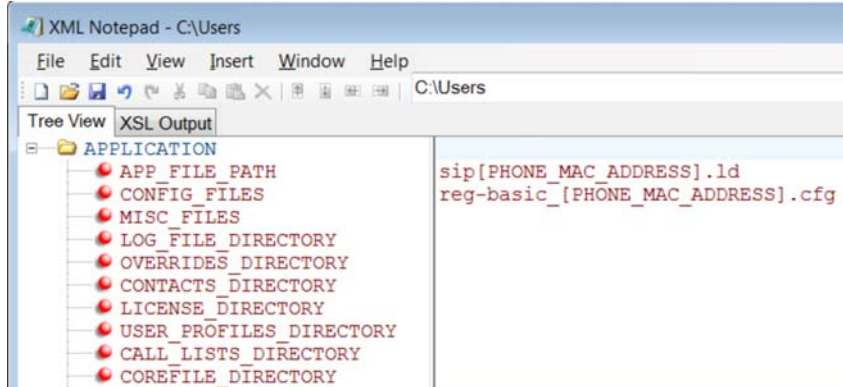
   4 Save the master configuration file.

Configure Phone Groups with the Master Configuration File

If you want to apply features or settings to a specific group of phones, you can create phone groups by:

- Using a variable to define per-phone configuration files.
- Appending a phone model name or part number to a configuration file.

Use a Variable in the Master Configuration File

You can use any of the following variable strings to create custom device groups:

- [PHONE_MODEL]
- [PHONE_PART_NUMBER]
- [PHONE_MAC_ADDRESS]

To get the model number or part number of a device, refer to System and Model Names.

To find a phone's MAC Address:

- **»** Find the MAC Address by doing one of the following:
  - ➢ Look on the label on the back of the phone
  - ➢ On the phone's menu system, go to **Settings** (**Menu** if using a VVX 1500) **> Status > Platform > Phone > S/N:**
  - ➢ Use a multikey shortcut by simultaneously pressing **1-4-7**

Apply Features and Settings to a Group of Phones

You can apply features and settings to a phone group by phone model name or part number. Note that if you create groups using the part number and model name for the same type of phone, the part number has priority over the model name, which has precedence over the original firmware version. The following is an example configuration for a group of VVX 500 business media phones.

To apply settings to a group of VVX 500 business media phones:

1 Create a configuration file with the settings you want to apply to the VVX 500 business media phones.

2 Rename the file using the part number or model name as a variable, for example:

   ➢ [3111-44500-001].cfg

   ➢ [VVX500].cfg

3 Add the file name to the CONFIG_FILES field of the master configuration file.

The following table lists the product name, model name, and part number for Polycom devices.

Product Name, Model Name, and Part Number

Product Name	Model Name	Part Number
SoundStructure VoIP Interface	SSTRVOIP	3111-33215-001
VVX D60 Wireless Handset	VVX D60	3111-17823-001
VVX 101	VVX 101	3111-40250-001
VVX 201	VVX 201	3111-40450-001
VVX 300	VVX 300	3111-46135-002
VVX 301	VVX 301	3111-48300-001
VVX 310	VVX 310	3111-46161-001
VVX 311	VVX 311	3111-48350-001
VVX 400	VVX 400	3111-46157-002
VVX 401	VVX 401	3111-48400-001
VVX 410	VVX 410	3111-46162-001
VVX 411	VVX 411	3111-48450-001
VVX 500	VVX 500	3111-44500-001
VVX 501	VVX 501	3111-48500-001
VVX 600	VVX 600	3111-44600-001
VVX 601	VVX 601	3111-48600-001
VVX 1500	VVX 1500	2345-17960-001

Configure Network Settings

The UC Software supports the deployment of Polycom phones for your device network:

- As a Session Initiation Protocol (SIP)-based endpoint interoperating with a SIP call server or softswitch. For more information on SIP, see the section Session Initiation Protocol (SIP).
- As an H.323 video endpoint (Polycom VVX 500/501, 600/601, and 1500 business media phones).

> **Web Info: Using VVX 1500 phones in a strict H.323 environment**
> For more information on using VVX 1500 phones in a strict H.323 environment, see *Deployment Guide for the Polycom VVX 1500D Business Media Phone* at Polycom VVX 1500 D on Polycom Support.

Polycom devices operate on an Ethernet local area network (LAN). Local area network design varies by organization and Polycom phones can be configured to accommodate a number of network designs. This section shows you several automated and manual ways to configure Polycom phones to operate on a LAN.

Connecting your Polycom phone to the LAN initiates the following startup sequence. Note the following:

- Only step 1 is required and automatic.
- Steps 2, 3, and 4 are optional as these settings can be manually configured on the device. It is common to complete step 3 using a DHCP server within the LAN.

Startup sequence:

1. The phone establishes network connectivity.

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

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

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

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

5. Provisioning server discovery.

   This is commonly done using DHCP as part of the previous step. As of UC Software 4.0, the phone contacts the provisioning server after the phone is operational in order to speed up boot time. You can disable the provisioning server discovery process as a way of reducing load on a provision server, for example, after a power failure.

Each step is explained in more detail in the following sections:

- Establish Phone Connection to the Network
- Apply Security and Quality of Service

- Configure Network Settings Using DHCPv4 or DHCPv6
- Provisioning Server Discovery

Establish Phone Connection to the Network

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

The phone supports two features to prevent Denial of Service (DoS):

- **Storm Filtering**   To change this parameter, see the section Modify Ethernet Settings.
- **VLAN Filtering**   To change this parameter, go to the section Modify VLAN Settings. VLAN filtering for the VVX business media phones is done by the Linux operating system and cannot be disabled.

Support for Storm and VLAN filtering varies by device.

Apply Security and Quality of Service

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.

Set Up VLANs and Wired Devices

You can use a virtual local area network (VLAN) to separate and assign higher priority to a voice LAN as a way of minimizing latency.

There are several methods you can use to configure the phone to work on a particular VLAN. If the phone receives a VLAN setting from more than one of the following methods, the priority is as follows (from highest to lowest):

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

Set Up 802.1X Authentication

802.1X authentication is a technology that originated for authenticating Wi-Fi links. It has also been adopted for authenticating computers within fixed LAN deployments. Multiple Device Authentication is available for Polycom devices as of UC Software 4.0.0.

Note that when VoIP phones with a secondary Ethernet port are used to connect computers on a network, the 802.1X authentication process becomes more complex since the computer is not directly connected to the 802.1X switch. To configure 802.1X, see the section Modify 802.1X Settings.

Web Info: Understand 802.1X

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 three 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). Note that the maximum number of individuals MAC addresses supported on the VVX embedded internal switch is 4,000. Note that the maximum number of individual MAC addresses supported on the VVX embedded internal switch is 4,000.

● Some switches support a feature that 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 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.

Configure Network Settings Using DHCPv4 or DHCPv6

After establishing network connectivity, the phone needs to acquire several IPv4 or IPv6 network settings. These settings are typically obtained automatically from a Dynamic Host Configuration Protocol (DHCPv4 or DHCPv6) server. You have the option to configure IPv4 or IPv6 network settings manually from the phone screen or using `device.set` capability. If you have never set up a provisioning server before, Polycom recommends reading the information provided in this section.

Web Info: RFC information on DHCP options

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

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

Timesaver: Reducing repetitive data

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

IPv4 Network Parameters

The following table lists the ways a phone can obtain IPv4 and related parameters in an IPv4 network.

IPv4 Network Parameters

Parameter	DHCPv4 Option	DHCPv4	DHCPv4 INFORM	Configuration File (application only)	*Device Settings*
IPv4 address	No	Yes	No	No	Yes
Subnet mask	1	Yes	No	No	Yes
IPv4 Gateway	3	Yes	No	No	Yes
Boot server address	Refer to Modify DHCP Settings or Provisioning Server Discovery.	Yes	Yes	No	Yes
SIP server address	151 **Note**: You can change this value by changing the device setting.	Yes	No	Yes	Yes
SNTP server address	Look at option 42, then option 4.	Yes	No	Yes	Yes
SNTP GMT offset	2	Yes	No	Yes	Yes
Syslog	Refer to the section Modify Syslog Settings	Yes	No	No	Yes
DNS server IP address	6	Yes	No	Yes	Yes
DNS INFORM server IP address	6	•	-	-	•
DNS domain	15	Yes	No	Yes	Yes
VLAN ID	Refer to the section Modify DHCP Settings	**Warning**: Link Layer Discovery Protocol (LLDP) overrides Cisco Discovery Protocol (CDP). CDP overrides Local FLASH which overrides DHCP VLAN Discovery.			

IPv6 Network Parameters

The following table lists the ways a phone can obtain IPv6 and related parameters in an IPv6 network.

IPv6 Network Parameters

Parameter	SLAAC[1]	DHCPv6 Option	DHCPv6	DHCPv6 INFORM	Configuration File (application only)	Device Settings
IPv6 Global Address	Yes	No	Yes	No	No	Yes
IPv6 ULA Address	Yes	No	Yes	No	No	Yes
IPv6 Gateway	Yes	No	No	No	No	Yes
Boot server IPv6 Address	No	Custom[2]	Yes	No	No	Yes
SIP server IPv6 Address	No	22/21	Yes	No	Yes	No
SNTP server IPv6 address	No	31	Yes	No	Yes	Yes
SNTP GMT offset	No	Custom[2]	Yes	No	Yes	Yes
Syslog IPv6 Address	No	Custom[2]	Yes	No	Yes	Yes
DNS server IPv6 address	No	23	Yes	No	Yes	Yes
IPv6 DNS domain	No	23	Yes	No	Yes	Yes
VLAN ID	Refer to the section Modify DHCP Settings	**Warning**: Link Layer Discovery Protocol (LLDP) overrides Cisco Discovery Protocol (CDP). CDP overrides Local FLASH which overrides DHCP VLAN Discovery.				

[1] Stateless Address Auto Configuration

[2] Polycom defined subcodes for these parameters as part of the DHCPv6 Vendor-Specific Information Option. Refer to section Polycom Vendor-Specific Information Options in DHCPv6.

Polycom Vendor-Specific Information Options in DHCPv6

In DHCPv6, there are no standard options defined for Boot Server IPv6 address, Syslog Server IPv6 Address, SNTP GMT Offset, and VLAN List. Polycom has defined subcodes for these specific information as part of the DHCPv6 Vendor-Specific Information Option. You can see the following sample configuration on a Linux DHCPv6 server for Polycom subcode definitions.

Sample configuration:

```
# Define PLCM options
```

```
option space plcm code width 2 length width 2 hash size 4;
option vsio.plcm code 13885 = encapsulate plcm;


option plcm.boot-server code 1 = string;
option plcm.time-offset code 2 = signed integer 32;
option plcm.syslog-server code 3 = string;
option plcm.vlan-list code 4 = string;


option plcm.boot-server "2620:0:1aa0:8071:d485:f47d:5de5:be04";
option plcm.time-offset 19850;
option plcm.syslog-server "2620:0:1aa0:8071:d485:f47d:5de5:be04";
option plcm.vlan-list "VLAN-A=513;";
```

> **Note: Overriding the DHCP value**
> You can configure parameter values for the **SNTP server address** and **SNTP GMT offset** to override the DHCP value. Refer to `tcpIpApp.sntp.address.overrideDHCP` for more information.
> The CDP Compatibility value can be obtained from a connected Ethernet switch if the switch supports CDP.

DHCP Network Parameters

The following table details the settings 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	Refer to Modify DHCP Settings or Provisioning Server Discovery.				
SIP server address	You can change this value by changing the device setting. Refer to <device/>.				
SNTP server address	Look at option 42, then option 4.				

DHCP Network Parameters

Parameter	DHCP Option	DHCP	DHCP INFORM	Configuration File (application only)	Device Settings
SNTP GMT offset	2				
Syslog	Refer to the section Modify Syslog Settings				
DNS server IP address	6				
DNS INFORM server IP address	6				
DNS domain	15				
VLAN ID	Refer to the section Modify DHCP Settings	**Warning**: Link Layer Discovery Protocol (LLDP) overrides Cisco Discovery Protocol (CDP). CDP overrides Local FLASH which overrides DHCP VLAN Discovery.			

DHCP Option 43

DHCP Option 60 controls how the phone identifies itself to a DHCP server for Polycom-specific options that must be returned. If the format for Option 60 is set to RFC 3925, then all Option 43 returned values are ignored. If the format for Option 60 is set to ASCII string, then the Option 43 response should have a hexadecimal string value encapsulating sub-options that override any options received outside of DHCP Option 43.

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. You can do this by connecting 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.

The following table lists supported DHCP Option 43 individual sub-options and combination sub-options.

DHCP Option 43 Configuration Options

Option	Results
Option 1- subnet mask	The phone parses the value from Option 43
Option 2 - Time offset	The phone parses the value.
Option 3 - Router	The phone parses the value.
Option 4 - TIME/ITP server address (RFC 868)	The phone parses the value.
Option 6 - Domain Name Server	The phone parses the value.
Option 7 - Domain Log server	The phone parses the value.
Option 15 - Domain Name	The phone parses the value.
Option 42 - Network Time Protocol server/SNTP server address (RFC 1769)	The phone parses the value.

DHCP Option 43 Configuration Options (continued)

Option	Results
Option 66 - TFTP Server Name	The phone parses the value.
Option 128 - 255	Available option range for configuring a custom boot server address when option 66 is not used.
Sub-options configured in Option 43	
Options 1, 2, 3, 4, 5, 6, 7, 15, 42, and 66	The phone parses the value.
Option 128 - 255	Available option range for configuring a custom boot server address when option 66 is not used.

Provisioning Server Discovery

After the phone has established network settings, the phone must discover a provisioning server that administrators typically use to obtain software updates and configuration settings. If you have never set up a provisioning server before, Polycom recommends reading the information provided in this section.

The phones support several methods to discover a 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 that takes the user 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 *Using Quick Setup with Polycom Phones: Technical Bulletin 45460 at* Polycom Engineering Advisories and Technical Notifications.

To change these parameters, refer to the section Modify Provisioning Server Settings.

Supported Provisioning Protocols

By default, phones are shipped with FTP enabled as the provisioning protocol. Note that there are two types of FTP method—active and passive—and UC Software is not compatible with active FTP. 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*. The server address can be an IP address, domain string name, or URL, and can be obtained through DHCP. For more information, refer to the section Modify Provisioning Server Settings.

Configuration file names in the ***<MACaddress>*.cfg** file can include a transfer protocol, for example, https://usr:pwd@server/dir/file.cfg. If you specify a user name and password as part of the server address or file name, they are 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.

The Updater performs the provisioning functions of uploading log files, master configuration files, software updates, and device setting menu changes. To guarantee software integrity, the Updater downloads only cryptographically signed Updater or UC Software images. Though UC Software supports digest and basic authentication when using HTTP/HTTPS, the Updater supports only digest authentication when using HTTP. When using HTTPS, the phone trusts widely recognized certificate authorities and you can add custom certificates to the phone. Note that log files are not appended when using TTP or HTTPS.

> **Settings: Choosing a valid URL**
>
> A URL must 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 are used.

> **Web Info: View trusted certificate authorities**
>
> For more information, see *Certificate Updates for Polycom UC Software* and *Using Custom Certificates with Polycom Phones: EA 17877* at Polycom Engineering Advisories and Technical Notifications.

As of SIP 3.2, TLS authentication is available. For more information, refer to the section Support Mutual TLS Authentication.

As of UC Software 4.0.0, 802.1X authentication is available. For more information, refer to the section Set Up 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.

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

Modify Phone Network Settings

You have the option to modify phone network settings. This section lists network settings available from the device interface. Use the soft keys, the arrow keys, and the Select and Delete keys to make changes.

If you have never set up a provisioning server before, Polycom recommends reading the information provided in this section.

You can update the network configuration parameters at one of two stages:

● **During the Updater Phase.** The setup menu is accessible during the auto-boot countdown of the Updater phase of operation. While your phone boots up, press the **Cancel** soft key, and press the **Setup** soft key to launch the setup menu. To access the setup menu, you must enter the administrator's password.

● **After your phone starts and is running UC Software.** The network configuration menu is accessible from the phone's main menu. Select **Menu > Settings > Advanced > Admin Settings > Network Configuration**. To access the **Advanced** menu, you must enter the administrator's password.

> **Tip: Changing the default administrator password**
> Polycom recommends that you change the default administrative password. Refer to the section Set Local User and Administrator Passwords.

Certain settings are read-only depending on the value of other settings. 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 is not used.

> **Tip: Resetting network configurations**
> The phone default network configuration referred to in subsequent sections can be reset to factory default settings using the phone's main menu: **Menu > Settings > Advanced > Admin Settings > Reset to Defaults > Reset Device Settings**. You can also use a multiple key combination, as shown in the section Define the Phone Key Layout.

Modify TR-069 Menu Settings

The following table describes TR-069 settings in the TR-069 Menu.

For details about this feature and instructions on how to set up TR-069, refer to Configure TR-069.

TR-069 Menu

Name	Permitted Values
Auto Configuration Server (ACS) Configuration	URL - Specify the URL of the ACS server.
	Username - Specify the TR-069 ACS server user name used to authenticate the phone.
	Password - Specify the TR-069 ACS server password used to authenticate the phone.
	Periodic Inform - Specify whether or not the customer-premises equipment (CPE) must periodically send CPE information to ACS using the Inform method call.
	Inform Interval - Specify the time interval in seconds in which the CPE must attempt to connect with the ACS to send CPE information if the Periodic Inform setting is enabled.

TR-069 Menu

Phone/CPE Configuration	Username - Specify the TR-069 CPE user name, which authenticates a connection request from the ACS server.
	Password - Specify the TR-069 CPE password, which authenticates a connection request from the ACS server.
Upgrade Management	Specify whether or not the ACS manages image upgrades for the phone.

Modify Main Menu Settings

You can modify the configuration settings shown in the following table from the setup menu while the phone boots, or from the phone Administrative Settings menu.

Main Menu

Name	Possible Values
Provisioning Menu	
Refer to the section Modify Provisioning Server Settings.	
Network Interfaces Menu or Ethernet Menu	
Refer to the Modify Ethernet Settings.	
TLS Security Menu	
Refer to the section Modify TLS Security Settings.	
SNTP Address	**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	**IP address**
The primary server the phone directs Domain Name System (DNS) queries to.	
DNS AltServer	**IP address**
The secondary server to which the phone directs DNS queries.	
DNS Domain	**Domain name string**
The phone's DNS domain.	

Main Menu

Hostname	**hostname**

The DHCP client hostname.

Syslog Menu

Refer to the section Modify Syslog Settings.

Quick Setup	**Enabled, Disabled**

If enabled, a QSetup soft key displays on the idle screen when you are in Lines View. When you press 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 Skype for Business-compatible phones to register with Skype for Business Server. When set to Lync, the phone automatically provisions with the minimum parameters required to register with Skype for Business Server. You cannot modify or customize the Base Profile.

By default, the Base Profile for normal SKUs is set to Generic.The Base Profile for Lync and Skype for Business SKUs is Lync.

Modify Provisioning Server Settings

You can modify the configuration settings shown in the table Provisioning Server Menu from the Provisioning Server menu on the phone.

Note: Change the server user and server password parameters
For security reasons, always change the Server User and Server Password fields from their default values.

Provisioning Server Menu

Name	Possible Values
DHCP Menu	

Refer to the section Modify DHCP Settings. 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.

Note: Active FTP is not supported for BootROM version 3.0 or later. Passive FTP is supported. Only implicit FTPS is supported.

Provisioning Server Menu

Server Address	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 uses 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. For information on how to specify a directory and use the master configuration file, see the section Configure with 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 is 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 is 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 is no further delay before the next attempt is started.

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 is accessed for software updates requested from the phone's Web Configuration Utility.

ZTP	Disabled, Enabled

See Zero-Touch Provisioning Solution on Polycom Support. Also see ZTP Frequently Asked Questions.

Modify DHCP Settings

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

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 selects the first response that contains a valid *custom* option value. If none of the responses contain a *custom* option value, the phone selects the first response that contains a valid *option66* value.

DHCP Menu

Name	Permitted Values
Boot Server	**Option 66, Custom, Static, Custom+Opt.66, DHCPv6, DHCPv6+Opt.66, DHCPv6+Custom, DHCPv6+Custom+Opt.66**

- **Option 66** The phone looks 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 the section Modify Provisioning Server Settings.
- **Custom** The phone looks 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 uses the boot server configured through the *Server Menu*. For more information, see the section Modify Provisioning Server Settings.
- **Custom + Option 66** The phone uses the custom option first or use Option 66 if the custom option is not present.
- **DHCPv6** This is applicable for IPv6 mode. The phone uses a DHCPv6 server to obtain provisioning server IPv6 address or FQDN.
- **DHCPv6+Opt.66** This is applicable for Dual Stack (IPv4 + IPv6) mode. The phone uses a DHCPv6 server to obtain provisioning server IPv6 address or FQDN. For provisioning server's IPv4 address, the phone uses Option 66.
- **DHCPv6+Custom** This is applicable for Dual Stack (IPv4 + IPv6) mode. The phone uses a DHCPv6 server to obtain provisioning server IPv6 address or FQDN. For provisioning server's IPv4 address, the phone uses Custom.
- **DHCPv6+Custom+Opt.66** This is applicable for Dual Stack (IPv4 + IPv6) mode. The phone uses a DHCPv6 server to obtain provisioning server IPv6 address or FQDN. For provisioning server's IPv4 address, the phone uses the custom option first or use Option 66 if the custom option is not present.

DHCP Menu

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 `0.0.0.0`, 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 retries and eventually times out.
- If the server address is not discovered using DHCP INFORM then the phone contacts the ZTP server if the ZTP feature is enabled. See ZTP Frequently Asked Questions.
- For DHCPv4, the phone sends out a DHCPv4 INFORM query.
  - ⌁ If a single DHCPv4 server responds to the INFORM query, this is functionally equivalent to the scenario where the primary DHCPv4 server responds with a valid boot server value or other DHCPv4 offered information.
  - ⌁ If no DHCPv4 INFORM server responds, the INFORM query process retries and eventually times out.
- If the server address is not discovered using DHCPv4 INFORM or DHCPv6, then the phone contacts the ZTP server if the ZTP feature is enabled. See ZTP Frequently Asked Questions.

Boot Server Option	**128 through 254 (Cannot be the same as VLAN ID Option)**

When the *Boot Server* parameter is set to Custom, this parameter specifies the DHCP option number in which the phone looks for its boot server.

Boot Server Option Type	**0=IP Address, 1=String**

When the *Boot Server* parameter is set to Custom, this parameter specifies the type of DHCP option in which the phone looks 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 the section Modify Provisioning Server Settings.

Option 60 Format	**0=RFC 3925 Binary, 1=ASCII String**

RFC 3925 Binary: Vendor-identifying information in the format defined in RFC 3925.
ASCII String: Vendor-identifying information in ASCII.
For more information, see *Using DHCP Vendor Identifying Options with Polycom Phones: Technical Bulletin 54041* at Polycom Engineering Advisories and Technical Notifications.

Note: DHCP option 125 containing the RFC 3295 formatted data is sent whenever option 60 is sent. DHCP option 43 data is ignored.

Modify Ethernet Settings

The Ethernet Menu is available only if there are multiple network interfaces to the phone.

Note: LAN port mode
You can set the LAN Port Mode on all phones. The PC Port Mode parameters are available only on phones with a second Ethernet port.
The following phones support the LAN Port Mode and PC Port Mode setting of 1000FD: VVX 310/311, 410/411, 500/501, 600/601, and 1500.

Configure Network Settings

Ethernet Menu

Name	Permitted Values
DHCP	**Enabled, Disabled**
If enabled, DHCP is used to obtain the parameters discussed in the section Modify DHCP Settings.	
IP Mode	**IPv4, IPv6, IPv4 and IPv6**
Specifies whether the phone network settings are configured for an IPv4, IPv6, or a dual stack (IPv4 and IPv6) network.	
Preferred Network	**IPv4, IPv6**
Specifies an IPv4 or IPv6 network as the preferred network to use in a dual stack environment.	
IP Address	**IP address**
The phone's IP address. Note: This parameter is disabled when DHCP is enabled.	
Subnet Mask	**Subnet mask**
The phone's subnet mask. Note: This parameter is disabled when DHCP is enabled.	
IP Gateway	**IP address**
The phone's default router.	
VLAN	
See the section Modify VLAN Settings.	
802.1X Authentication	**Enabled, Disabled**
If enabled, the phone uses the 802.1 Authentication parameters to satisfy the negotiation requirements for each EAP type.	
802.1X Menu	
See the section Modify 802.1X Settings.	
Storm Filtering	**Enabled, Disabled**
If enabled, received Ethernet packets are filtered so that the TCP/IP stack does not process bad data or too much data. The default value is Enabled.	
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.	

Ethernet Menu

PC Port Mode	0 = Auto, 1 = 10HD, 2 = 10FD, 3 = 100HD, 4 = 100FD, 5 = 1000FD, -1 = Disabled

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.

IPv4 Configuration Menu

The following table lists IPv4 options you can configure from the phone menu.

IPv4 Options

Name	Permitted Values
DHCP	**Enabled, Disabled**
If enabled, DHCP is used to obtain the parameters discussed in the section Modify DHCP Settings.	
IP Address	**IPv4 Address**
The phone's IPv4 address. Note: This parameter is disabled when DHCP is enabled.	
Subnet Mask	**Subnet Mask**
The phone's subnet mask for IPv4 network. Note: This parameter is disabled when DHCP is enabled.	
IP Gateway	**IPv4 address**
The phone's default gateway router.	

IPv6 Configuration Menu

The following table lists IPv6 options you can configure from the phone menu.

IPv6 Options

Name	Permitted Values
IPv6 Addr Discovery	**Static, DHCPv6, SLAAC**
The IPv6 addresses can be configured statically or through dynamic address configuration with a DHCPv6 server or through stateless auto address configuration with SLAAC.	
Global Addr	**IPv6 Address**
The phone's global IPv6 address. This is enabled only if IPv6 Address Discovery is static.	
Link Local Addr	**IPv6 Address**
The phone's link local IPv6 address. It is auto-generated by default. You manually provide a link local address, which would override the auto-generated address.	
ULA	**IPv6 Address**

IPv6 Options

Name	Permitted Values
The phone's unique local IPv6 address. This is enabled only if IPv6 Address Discovery is static.	
IPv6 Gateway	**IPv6 Address**
The default IPv6 gateway address. If it is not manually provided, then it is obtained through Router Advertisements.	

Modify VLAN Settings

You can modify the settings listed in the following table.

VLAN Menu

Name	Permitted Values
IPv6 VLAN Discovery	**Enabled, Disabled**
If enabled, the VLAN IDs are obtained from a DHCPv6 server. This parameter applies only if the phone is in IPv6 or Dual Stack mode.	
VLAN ID	**Null, 0 through 4094**
The phone's 802.1Q VLAN identifier. The default value is Null. Note: Null = no VLAN tagging	
LLDP	**Enabled, Disabled**
If enabled, the phone uses the LLDP protocol to communicate with the network switch for certain network parameters. Most often this is 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, refer to the section LLDP and Supported TLVs.	
CDP Compatibility	**Enabled, Disabled**
If enabled, the phone uses CDP-compatible signaling to communicate with the network switch for certain network parameters. Most often this is 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.	
VLAN Discovery	**0=Disabled, 1=Fixed (default), 2=Custom**
• **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 is ignored**.** • Custom: Use the number specified for VLAN ID Option as the DHCP private option value. For a detailed description, refer to the section Assign a VLAN ID Using DHCP.	
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 a detailed description, refer to the section Assign a VLAN ID Using DHCP.	

Modify 802.1X Settings

The 802.1X Menu displays on the VVX 500/501 and 600/601 when 802.1X authentication is enabled.

You can modify configuration parameters shown in the following table.

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 the section 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 the section Modify PAC File Settings.	
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.	

Modify PAC File Settings

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

PAC File Information Menu

Name	Possible Values
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.	

PAC File Information Menu

Name	Possible Values
Remove PAC File	**UTF-8 encoded string**
A flag to determine whether or not to delete the PAC file from the phone.	

Modify Login Credential Settings

You can modify settings shown in the following table.

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.	

Modify TLS Security Settings

This section refers to the TLS Security menu available in the Updater and UC Software. You can modify the settings shown in the following table.

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.	
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.	

TLS Security Menu

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 the section Modify TLS Profile Settings.	
Web Services	
See the section Modify Applications Settings.	

Modify TLS Profile Settings

You can modify settings shown in the following table.

TLS Profile Menu

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

Modify Applications Settings

You can modify settings shown in the following table.

Applications Menu

Name	Possible Values
802.1X The TLS Profile to use for 802.1X authentication.	**1 or 2**
Provisioning The TLS Profile to use for provisioning authentication.	**1 or 2**
Provisioning	**Enable or Disable**

Applications Menu

The TLS Profile to enable or disable common name validation.

Syslog	**1 or 2**

The TLS Profile to use for Syslog authentication.

Modify Syslog Settings

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 settings shown in the following table.

Syslog Menu

Name	Possible Values
Server Address	**IP address or domain name string**
The syslog server IP address. The default value is Null.	
Server Type	**None=0, UDP=1, TCP=2, TLS=3**
The protocol that the phone uses 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 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 Addres	**Enabled, Disabled**
If enabled, the phone's MAC address is prepended to the log message sent to the syslog server.	

Configure Devices and Call Controls

This section provides information to successfully configure the UC Software configuration files and parameters that controls device features and settings.

How to Read the Feature Parameter Tables

Each of the feature descriptions includes a table of parameters that you configure to make the features work. The following section explains the conventions used in the feature parameter tables.

The feature parameter tables indicate the parameters you can configure for a feature when using the centralized provisioning method. The following illustration shows you how to read the feature parameter tables to locate the function of the parameter, the name of the template file the parameter is in, and the name of the parameter you configure. Each parameter listed in the feature parameter tables is linked to the reference section Configuration Parameters in this guide, which provides a full description, permissible, and default values for each parameter.

Feature parameter table format

Parameter Function	Template Name	Parameter Name
Parameter Function	**template** > parameter	
Filter RTP packets by IP address.	**site.cfg** > tcpIpApp.port.rtp.filterByIp	
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	

To locate a parameter in your UC Software download, 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 following example is taken from the section Call Forward on Shared Lines.

Feature parameter table for multiple call appearances

Parameter Function	template > parameter
Set the default number of concurrent calls for all line keys	**reg-basic.cfg** > call.callsPerLineKey

This example indicates that the **reg-basic.cfg** template file contains the `calls.callsPerLineKey` parameter, which sets the number of concurrent calls for a phone's line keys. Because the default value varies by device, click on the blue parameter name to go to the parameter description, which includes default and permissible values. If you want to change the parameter value, locate and open the reg.basic template, expand the reg folder, and locate the parameter `calls.callsPerLineKey`. Set the parameter value to a permissible number, as shown in the following illustration.

Example Multiple Call Appearances
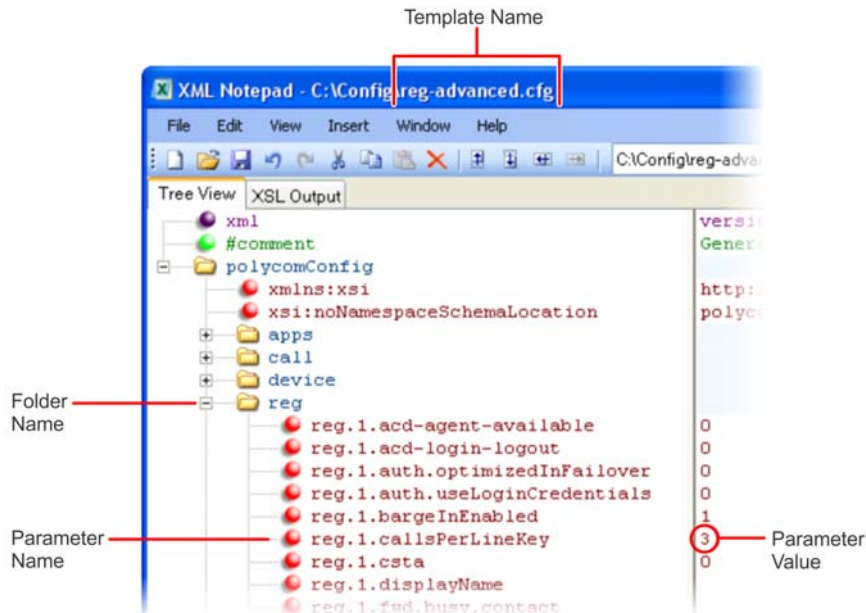
Some file paths in the templates are long and you may have to expand several folders in the XML tree structure to locate a specific parameter.

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

Note: Each parameter is linked

Each parameter listed in the tables in various sections is linked to its definition in the section Configuration Parameters. This section defines 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 blue parameter name.

Example Two: Configuring Grouped Parameters

Some of the features have several related parameters that you need to configure to get the feature working. In these cases, instead of listing every parameter, the table specifies 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 Configuration Parameters. 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

Parameter Function	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
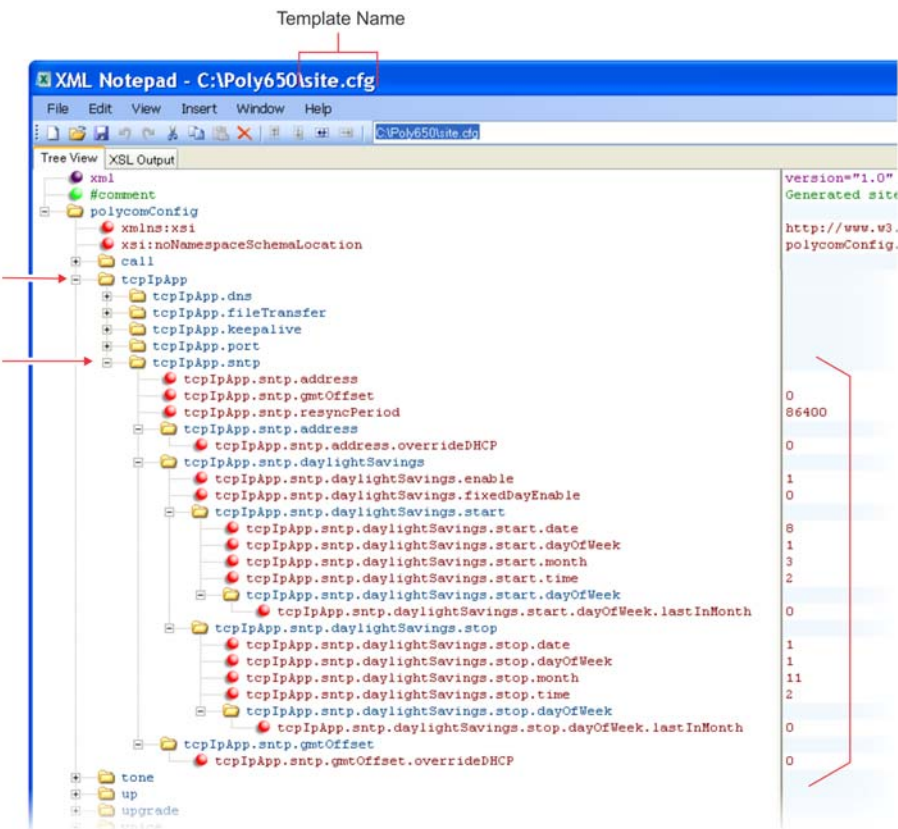
Note: Using an XML editor

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

Configure Phone Signaling

This section provides information on configuring phone signaling.

Support for 3GPP Technical Specifications

For an IP Multimedia Subsystem (IMS) environment, Polycom has introduced support for a subset of the 3rd Generation Partnership Project technical specifications (3GPP TS) 24.229, 24.615, and 24.629. In addition, Polycom phones provide partial or complete support for the following RFCs:

- RFC 3327
- RFC 3608
- RFC 3680
- RFC 6665
- RFC 6228
- RFC 3261
- RFC 5009
- RFC 7462
- RFC 7329
- RFC 6026
- RFC 3581
- RFC 6947

This release adds the following IMS feature enhancements:

- The call waiting ringback tone plays to inform you that the call is waiting at the far end.

- The SIP Response Code 199 (defined in RFC 6228) is supported.

- The Path extension header field in the SIP Register request message allows accumulating and transmitting the list of proxies between a user agent and Registrar server. You can configure the parameter `reg.x.path` to enable or disable support for this header field for a specific line registration.

- The caller phone can support the p-early-media SIP header that determines whether the caller phone should play a network-provided media or its own media as a ring back tone. Configure the parameter `voIpProt.SIP.header.pEarlyMedia.support` to enable or disable support for this header field on the caller phone.

- The VQMon messages that are generated by the phone can contain service route information in SIP route headers. Configure the parameter `voice.qualityMonitoring.processServiceRoute.enable` to enable or disable this header field on the VQMon messages generated by a phone device.

- In a NAT network, a phone may need to send keep-alive messages to maintain the IP addresses mapping in the NAT table. The parameters `nat.keepalive.udp.payload` and `nat.keepalive.tcp.payload` configure a customizable string as the payload of the UDP and TCP keep-alive messages.

The following table lists available 3GPP parameters you can configure.

3GPP Parameters

Parameter Template	Permitted Values
nat.keepalive.tcp.payload features.cfg	Configure a customizable string as the payload of a TCP keep-alive message. The string value cannot be blank. CRLFCRLFCRLFCRLFCRLFCRLFCRLFCRLF (default)
nat.keepalive.udp.payload features.cfg	Configure a customizable string as the payload of a UDP keep-alive message. You can leave the string value blank to configure an empty payload. CRLFCRLF (default)
reg.x.header.pearlymedia.support features.cfg	0 (Default) - The p-early-media header is not supported on the specified line registration. 1 - The p-early-media header is supported by the specified line registration.
reg.X.insertOBPAddressInRoute reg-basic.cfg	1 (Default) - The outbound proxy address is added as the topmost route header. 0 - The outbound proxy address is not added to the route header.
reg.x.path features.cfg	0 (Default) - The path extension header field in the Register request message is not supported for the specific line registration. 1 - The phone supports and provides the path extension header field in the Register request message for the specific line registration.
reg.x.regevent features.cfg	0 (default) - The phone is not subscribed to registration state change notifications for the specific phone line. 1 - The phone is subscribed to registration state change notifications for the specific phone line. This parameter overrides the global parameter voIpProt.SIP.regevent.
reg.x.rejectNDUBInvite features.cfg	Specify whether or not the phone accepts a call for a particular registration in case of a Network Determined User Busy (NDUB) event advertised by the SIP server. 0 (Default) - If an NDUB event occurs, the phone does not reject the call. 1 - If an NDUB event occurs, the phone rejects the call with a 603 Decline response code.

3GPP Parameters

Parameter Template	Permitted Values
`reg.x.server.y.specialInterop` features.cfg	Specify the server-specific feature set for the line registration. Standard (Default) VVX 101: Standard GENBAND GENBAND-A2 ALU-CTS DT VVX 201: Standard, GENBAND GENBAND-A2 ALU-CTS ocs2007r2 lync2010 All other phones: Standard GENBAND GENBAND-A2 ALU-CTS ocs2007r2 lync2010 lcs2005
`voice.qualityMonitoring.processServiceRoute.enable` features.cfg	0 (Default) - The VQMon messages generated by the phone do not contain service route information in SIP route headers. 1 - The VQMon messages generated by the phone contain service route information, if available, in SIP route headers.
`voIpProt.SIP.header.pEarlyMedia.support` sip-interop.cfg	0 (Default) - The p-early-media header is not supported by the caller phone. 1 - The p-early-media header is supported by the caller phone.
`voIpProt.SIP.IMS.enable` features.cfg	This parameter applies to all registered or unregistered SIP lines on the phone. 0 (Default) - The phone does not support IMS features introduced in UC Software 5.5.0. 1 - The phone supports IMS features introduced in UC Software 5.5.0.

3GPP Parameters

Parameter Template	Permitted Values
`voIpProt.SIP.regevent` features.cfg	0 (default) - The phone is not subscribed to registration state change notifications for all phone lines. 1 - The phone is subscribed to registration state change notifications for all phone lines. This parameter is overridden by the per-phone parameter `reg.x.regevent`.
`voIpProt.SIP.rejectNDUBInvite` features.cfg	Specify whether or not the phone accepts a call for all registrations in case of a Network Determined User Busy (NDUB) event advertised by the SIP server. 0 (Default) - If an NDUB event occurs, the phone does not reject the call for all line registrations. 1 - If an NDUB event occurs, the phone rejects the call with a 603 Decline response code for all line registrations.
`voIpProt.SIP.supportFor199` reg-basic.cfg	Determine support for the 199 response code. For details on the 199 response code, see RFC 6228. 0 (Default) - The phone does not support the 199 response code. 1- The phone supports the 199 response code.

Advice of Charge

In an IP Multimedia Subsystem (IMS) environment, Polycom phones support the Advice of Charge (AoC) feature as defined in Technical Specification (TS) 24.647 version 9.1.0 Release 9.

You can enable Polycom phones to display call charges information which can include:

- Call setup charge and call tariff information – Displayed at the beginning of a call.
- Cumulative call cost – Displayed on an ongoing call.
- Complete call cost – Displayed after a call ends.

Configure Advice of Charge

The following parameters configure the Advice of Charge (AoS) feature. Before configuring AoS parameters, you must set `voIpProt.SIP.IMS.enable=1`.

Advice of Charge Parameters

Parameter template	Permitted Values
`voIpProt.SIP.aoc.enable` sip-interop.cfg	0 (Default) - The phone does not display call charge information. 1 - The phone displays call charge information.
`feature.adviceOfCharge.allowAudioNotification` features.cfg	0 (Default) - There is no audio beep sound when the call charges information is updated on the phone display. 1 - The phone gives an audio beep when the call charges information is updated on the phone display.

IPv6 Protocol Support

VVX business media phones support IPv6 and you can configure the phones to operate in IPv4, IPv6, or dual stack (IPv4/IPv6) mode.

You can enable and configure IPv6 support from the phone menu, the Web Configuration Utility, or with centralized provisioning.

Configure IPv6 with Centralized Provisioning

The following table lists parameters you can use to enable and configure IPv6.

IPv6 Parameters

Parameter Template	Permitted Values
`device.dhcp.bootSrvUseOpt` site.cfg	Specifies the source for the boot server address for the phone. It can take values from 0 to 9. In IPv4 mode, the following values are applicable: • 0 (Default) - The phone gets the boot server address from Option 66. • 1 - The phone gets the boot server details from the custom option number provided through DHCP. • 2 - The phone uses the boot server configured through the Server Menu. • 3 - The phone uses the custom option first or uses Option 66 if the custom option is not present In IPv6 mode, the following values are applicable: • 4 - The phone uses the boot server configured through the Server menu. • 5 - The phone uses the boot server option provided through DHCPv6. In Dual Stack Mode (IPv4/IPv6 mode), the following values are applicable: • 6 - The phone uses the boot server configured through the Server menu. • 7 - The phone gets the boot server details from the DHCPv6 option or Option 66 on the DHCP server. • 8 - The phone gets the boot server details through DHCPv6 or through the custom option configured on the DHCP server. • 9 - The phone gets the boot server from the DHCPv6 option, custom option, or Option 66 configured on the DHCP server.
`device.net.ipStack` site.cfg	Configures the IPv4, IPv6, or dual stack mode for the phone. 0 (Default) - IPv4 is enabled and IPv6 is disabled. 1 - IPv6 is enabled and IPv4 is disabled. 2 - Dual stack is enabled and phone operates on both IPv4 and IPv6 networks at the same time.
`device.net.ipv6AddrDisc` site.cfg	Specify whether the IPv6 address and related parameters for the phone are obtained from DHCPv6 or SLAAC or statically configured for the phone. 1 (Default) -IPv6 global address and options are configured from DHCPv6. 2 - IPv6 global address is configured using prefixes received from Router Advertisements (RA) and options are configured from stateless DHCPv6. 0 - IPv6 global address and options must be configured manually.
`device.net.ipv6Address` site.cfg	Specify a valid global IPv6 unicast address for the phone. Null (default)

IPv6 Parameters

Parameter Template	Permitted Values
`device.net.ipv6Gateway` site.cfg	Specify theIPv6 address of the default gateway for the phone. Null (default)
`device.net.ipv6LinkAddress` site.cfg	Specifies a valid Link Local IPv6 address for the phone. Null (default)
`device.net.ipv6PrivacyExten sion` site.cfg	Configure whether or not the IPv6 global and link local addresses are in 64-bit Extended Unique Identifier (EUI-64) format. 0 (Default) - IPv6 global and link local addresses are in EUI-64 format. 1 - Global and link local IPv6 addresses are not in EUI-64 format. Instead, the last 48 bits for the IPv6 address are generated randomly.
`device.net.ipv6ULAAddress` site.cfg	Specifies a valid Unique Local IPv6 address (ULA) for the phone. Null (default)
`device.net.preferredNetwork` site.cfg	Specify IPv4 or IPv6 as the preferred network in a Dual Stack mode. 1 (default) - Specifies IPv6 as a preferred network. 0 - Specifies IPv4 as a preferred network.
`voipProt.SIP.anat.enabled` sip-interop.cfg	Enables or disables Alternative Network Address Types (ANAT). 0 (default) - ANAT is disabled. 1 - ANAT is enabled.

Configure Real-Time Transport Protocol (RTP) 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 following table 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 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

Parameter Function	**template** > parameter
Filter RTP packets by IP address.	**site.cfg** > tcpIpApp.port.rtp.filterByIp
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

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 rejects 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 are not rejected. Enter a value in the `tcpIpApp.port.rtp.forceSend` parameter to specify the port that all RTP packets are 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.

Default real-time transport protocol

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 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 following table 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

Parameter Function	**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

Network Access Translation

Specify the external NAT media port start.	**sip-interop.cfg** > nat.mediaPortStart
Specify the external NAT signaling port.	**sip-interop.cfg** > nat.signalPort

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 is set the initially allocated RTP port and overrides the value set in `tcpIpApp.port.rtp.mediaPortRangeStart`. In the example, the starting port is 12500 and the phone cycles 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 uses for SIP signaling. This parameter overrides `voIpProt.local.Port`. In the example below, the phone uses 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 sends 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 sends the keepalive every 120 seconds.

Default NAT parameter settings
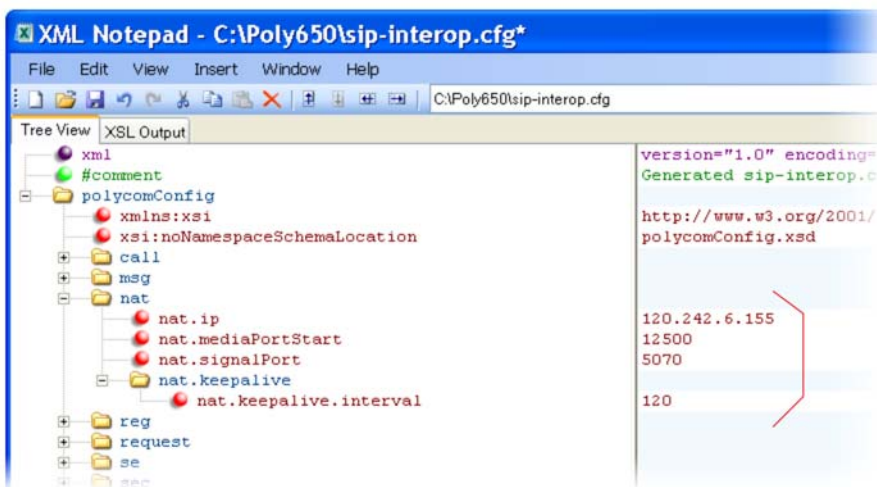
DNS SIP Server Name Resolution

If a DNS name is given for a proxy/registrar address, the IP addresses associated with that name is discovered as specified in RFC3263. If a port is given, the only lookup is an A record. If no port is given, NAPTR and SRV records are 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, port 5060 is used. If the registration type is TLS, port 5061 is used.

Caution: No DNS resolution causes failover

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

The following configuration causes the phone to build an SRV request based on the address you provide, including all subdomains. Use the format:

- voIpProt.SIP.outboundProxy.address="<*sip.example.com>*"
- voIpProt.SIP.outboundProxy.port="0"

This SRV request produces a list of servers ordered by weight and priority, enabling you to specify subdomains for separate servers, or you can create partitions of the same system. Please note that while making SRV queries and transport is configured as TCP, the phone adds the prefix `<_service._proto.>` to the configured address/FQDN but does not remove the subdomain prefix, for example `sip.example.com` becomes `_sip._tcp.sip.example.com`. A single SRV query can be resolved into many different servers, session border controllers (SBCs), or partitions ordered by weight and priority, for example, `voice.sip.example.com` and ***video.sip.example.com***. Alternatively, use DNS NAPTR to discover what services are available at the root domain.

Behavior When the Primary Server Connection Fails

For Outgoing Calls (INVITE Fallback)

When the user initiates a call, the phone completes the following steps to connect the call:

1 The phone tries to call the working server.

2 If the working server does not respond correctly to the INVITE, the phone tries and makes 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 tries 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 fails.

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 attempts to contact the DNS server to resolve the address of all servers in its list before initiating a call. These attempts 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) is configured with the address of the service provider call server. The IP address of the server(s) is provided by the DNS server, for example: `reg.1.server.1.address=voipserver.serviceprovider.com`.
- Server 2 (the fallback server) is 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 ensure that the phone and network load generated by registration refresh of multiple registrations does not become excessive. This is of particular concern when a phone has multiple registrations with multiple servers per registration and some of these servers are unavailable.

Phone Operation for Registration

After the phone has booted up, it registers to all configured servers.

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, presence, and BLF is established only with Server 1.

Upon the registration timer expiry of each server registration, the phone attempts to re-register. If this is unsuccessful, normal SIP re-registration behavior (typically at intervals of 30 to 60 seconds) proceeds and continues until the registration is successful (for example, when the Internet link is again operational). While the primary server registration is unavailable, the next highest priority server in the list serves as the working server. As soon as the primary server registration succeeds, it returns 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 does not register to that server. However, the INVITE fails 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.
- Avoid using too many servers as part of the redundancy configuration as each registration generates more traffic.
- Educate users as to the features that are not available when in fallback operating mode.

> **Note: Compatibility with Microsoft Skype for Business**
> The concurrent/registration failover/fallback feature is not compatible with Microsoft environments.

Configure the Static DNS Cache

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.

Phones configured with a DNS server behave by default as follows:

- The phone makes an initial attempt to resolve a hostname that is within the static DNS cache. For example, a query is 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 again results 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.

Phones not configured with a DNS server behave by default as follows:

- When the phone attempts to resolve a hostname within the static DNS cache, it always returns 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 RFC2308.

Configuring the Static DNS Cache

Parameter Function	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.	**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.*
Specify whether to use DNS primary and secondary address set using the parameters tcpIpApp.dns.server and tcpIpApp.dns.altServer.	**site.cfg** > tcpIpApp.dns.address.overrideDHCP
Specify whether to use the DNS domain name set using the parameter tcpIpApp.dns.domain.	**site.cfg** > tcpIpApp.dns.domain.overrideDHCP

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

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 looks 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 looks 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: Order of addresses**
> The addresses listed in this example are read by Polycom UC Software in the order listed.

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 looks 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 looks 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	

Note: 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 not used, the **site.cfg** configuration looks 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 looks as follows:

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.

Enable Access URL in SIP Messages

When this feature is enabled, the server can attach a URL to incoming and active calls. The phone's browser or microbrowser can read this URL and render it as web content that displays on the phone screen. This feature can be enabled on VVX 500/501 and 1500 phones as shown in the table Enable Access URL in SIP Messages.

This feature is flexible and can be used in the following ways.

- A Call Center
  - ➢ A URL is attached to an incoming call and displays extended information about a customer before the agent takes the call.
  - ➢ The phone can display a script of questions for an agent to ask a caller, and a different script can be provided to different agent groups.
- A Restaurant menu on a hotel phone
  - ➢ A guest dials a number for the restaurant or room service and a voice indicates that the menu is available for viewing on the phone.

There are three user interface aspects to this feature:

- **Web Content Status Indication**—When valid web content is available on the phone, an icon displays beside the call information. In the examples shown next, a lightning bolt icon indicates web content is available for a call appearance. The phone user can press the Select key to display the web content.

- **Web Content Retrieval**—Phone users can choose to retrieve web content in Active Mode (spontaneously) or in Passive Mode (by request).
  - ➢ **Active Mode**—There are two ways to configure spontaneous web content retrieval: you can set the web content retrieval parameter in the configuration file to 'active' or, if your call server supports access URL, you can specify active retrieval in the SIP heading. If parameters in the SIP signal conflict with the file configuration, parameters in the SIP signaling takes precedence. Note that incoming active web content interrupts web content currently being viewed.
  - ➢ **Passive Mode**—There are two ways to configure web content retrieval by request: you can set the web content retrieval parameter in the configuration file to 'passive' or, if your call server supports access URL, you can specify passive retrieval in the SIP heading. When passive mode is enabled, an icon displays beside a call appearance indicating that web content is available. For more information about the web content icon, see Web Content Status Indication earlier in this section. When an icon shows that web content is available, the phone user can press the Select key to view the content. If the web content has expired, no icon displays and the Select key performs no function. Note that incoming active web content interrupts web content currently being viewed. Passive mode is recommended when the microbrowser is used for other applications.
- **Settings Menu**—You can enable new web content to be added to the phone's menu. Using the phone's menu, users can set the default display mode for individual URLs to active or passive.

You must use the following standards if you want to set the retrieval display mode of web content in the SIP headers:

- ➢ A new SIP header must be used to report web content associated with SIP phone calls (the SSAWC header follows the BNF for the standard SIP header Alert-Info):

```
Alert-Info = "Alert-Info" HCOLON alert-param *(COMMA alert-param)

alert-param = LAQUOT absoluteURI RAQUOT *( SEMI generic-param )
```

The web content must be located with an absolute URI that begins with the scheme identifier. Currently only the HTTP scheme is supported.

The following is an example of a valid SIP header:

```
Access-URL: <http://server.polycom.com/content23456.xhtml>
```

This header may be placed in SIP requests and responses so long as the messages are part of an INVITE-initiated dialog and the phone can associate them with an existing phone call.

You may also define two optional parameters:

- ➢ An `expires` parameter is defined to indicate the lifespan of the URL itself. Or, if the URL is permanent, you can set how long the web content displays with the call. An absent or invalid parameter is interpreted to mean that the content or the URL itself is persistent. A value, if present, indicates the lifespan of the content in seconds (zero has special significance—see the next example). When the lifespan expires, the phone removes both the indication of the URL and the ability of the user to retrieve it.

  For example:

```
Access-URL: <http://server.polycom.com/content23456.xhtml>; expires=60
```

  If the server wishes to invalidate a previous URL, it can send a new header (through UPDATE) with expires=0. The expires parameter is ignored when determining whether to spontaneously retrieve the web content unless expires=0.

- ➢ A mode parameter is defined to indicate whether the web content should be displayed spontaneously or retrieved on-demand. Two values are allowed: active and passive. An absent or invalid parameter is interpreted the same as passive, meaning that the web content is

retrievable on-demand but is not be spontaneously displayed. If the value is set to active, the web content is spontaneously displayed, subject to the rules discussed under Active Mode in Web Content Retrieval earlier in this section.

For example:

```
Access-URL: <http://server.polycom.com/content23456.xhtml>;expires=60;
mode=passive
```

In this case, an icon indicates that web content is available for a period of 60 seconds.

Enable Access URL in SIP Messages

Parameter Function	**template** > parameter
To turn this feature on or off.	**features.cfg** > mb.ssawc.enabled
To retrieve content.	**features.cfg** > mb.ssawc.call.mode

Example Access URL in SIP Messages Configuration

In the following example, in the **features.cfg** template, the access URL in SIP message feature is enabled in mb.ssawc.enabled. The parameter mb.ssawc.call.mode is set to passive, which means web content does not display spontaneously; web content displays when activated by the phone user.

Access URL in SIP messages example

Display 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, refer to the section Supported SIP Request Headers.

You can display these pop-up messages in any language supported by the phone. The messages 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 the following table.

SIP Header Warnings

Parameter Function	**template** > parameter
Turn this feature on or off.	**sip-interop.cfg** > voIpProt.SIP.header.warning.enable
Specify which warnings can be displayed.	**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:
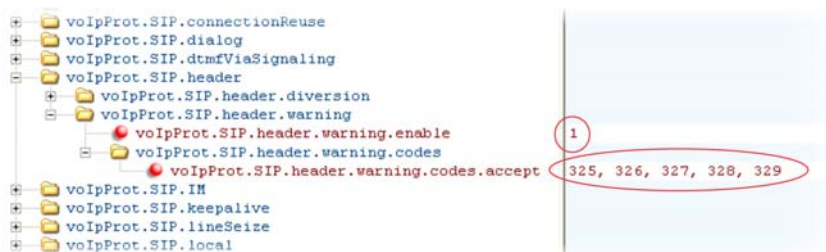
Set Up Server Redundancy

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 lists 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.

> **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 only supports advanced SIP features such as shared lines, missed calls, and presence with the primary server (y=1).

● **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). Polycom phones support 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 environments**
> The concurrent failover/fallback feature is not compatible with Microsoft environments.

Set Up Server Redundancy

Parameter Function	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.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Web Info: Failover configuration details

For more information, see *Technical Bulletin 5844: SIP Server Fallback Enhancements on Polycom Phones* and *Technical Bulletin 66546: Configuring Optional Re-Registration on Failover Behavior*.

Configure Presence

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.

VVX phones support a maximum of 64 buddies for Open SIP server platforms and 200 contacts on Skype for Business server. For information on the Skype for Business contacts, refer to the *Polycom UC Software with Microsoft Lync Server and Skype for Business - Deployment Guide* on Polycom Voice Support.

The following table lists the parameters you can configure. Note that other phone users can block you from monitoring their phones.

For more information about the Microsoft Skype for Business or BroadSoft UC-One presence features, see the *Polycom VVX Business Media Phones User Guide.*

Use the Presence Feature

Parameter Function	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

Example Presence Configuration

In the following illustration, the presence feature has been enabled in `feature.presence.enabled`. The **MyStatus** and **Buddies** soft keys both display on the phone's home screen when you enable the `pres.idleSoftkeys` parameter. The `pres.reg` parameter uses the address of phone line 1 for the presence feature.
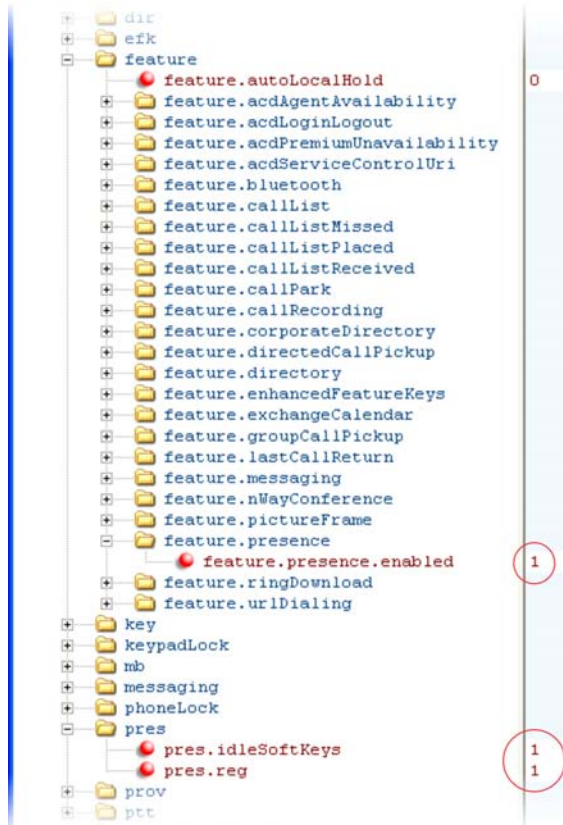
This configuration enables the presence feature and display the **MyStatus** and **Buddies** soft keys on the phone. When you press the Buddies soft key, contacts you have entered to your buddy list displays.

Provisional Polling of Polycom Phones

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

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 polls 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, polls 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 Polycom Phones

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

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

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 polls once per day, between 1 and 5 am.
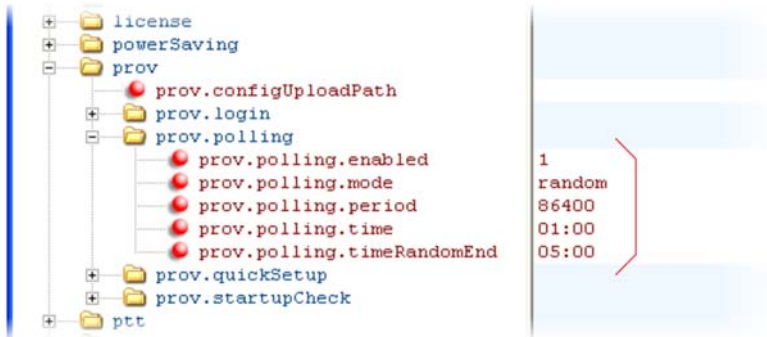
Note: Only provision files when polling

If `prov.startupCheck.enabled`='0' then Polycom phones do not look for the sip.ld or the configuration files when they are rebooted, lose power, or restarted. Instead, they 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 are downloaded each time as they are stored in RAM and lost with every reboot.

Configure SIP Subscription Timers

This feature enables you to configure a subscription expiry independently of the registration expiry. You can also configure an overlap period for subscription independently of the overlap period for the registration, and

a subscription expiry and subscription overlap for global SIP servers and per-registration SIP servers. Note that per-registration configuration parameters override global parameters. If you have not explicitly configured values for any user features, the default subscription values are used.

SIP Subscription Timers

Parameter Function	template > parameter
A global parameter that sets the phone's requested subscription period.	**reg-advanced.cfg** > voIpProt.server.x.subscribe.expires
A global parameter that sets the number of seconds before the expiration time returned by server x after which the phone attempts to resubscribe.	**reg-advanced-cfg** > voIpProt.server.x.subscribe.expires.overlap
A per-registration parameter that sets the phone's requested subscription period.	**reg-advanced-cfg** > reg.x.server.y.subscribe.expires
A per-registration parameter that sets the number of seconds before the expiration time returned by server x after which the phone attempts to resubscribe.	**reg-advanced-cfg** > reg.x.server.y.subscribe.expires.overlap

Default Features

This section lists phone features for which no configuration is required.

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.

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 displays caller IDs matched to the call server and does not match IDs to entries in the local contact directory or corporate directory.

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 logs and displays 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.

Microphone Mute

All phones have a microphone mute button. When you activate microphone mute, a red LED glows or a mute icon displays on the phone screen, depending on the phone model you are using.

No configuration changes can be made to the microphone mute feature.

Automatic Gain Control

Automatic Gain Control (AGC) is applicable to conference phone models and is used to boost the transmit gain of the local talker in certain circumstances. This increases the effective user-phone radius and helps you to hear all participants equally.

Background Noise Suppression

Background noise suppression is designed primarily for handsfree operation and reduces background noise, such as from fans, projectors, or air conditioners, to enhance communication.

Synthesized Comfort Noise Fill

This feature is an integral part of handsfree echo reduction; it is unrelated to Comfort Noise packets generated if Voice Activity Detection is enabled. Synthesized 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.

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.

Configure Phone Alerts

This section shows you how to configure phone and call alert features.

Enable Persistent Mute

This feature enables you to have the mute state of your phone persist across calls. Default mute behavior allows you to activate the mute state only if the phone is in an active call and ends when the active call ends. When you enable this feature and press Mute, the phone stays in the mute state until you press Mute again or until the phone restarts. When you mute the phone in an idle state, the mute LED glows but no icon displays on the screen. When you initiate a new active call with mute on, the mute LED glows and a Mute icon displays on the phone screen.

Persistent Mute Parameters

Parameter Function	template > parameter
Enable or disable the persistent mute feature.	**features.cfg** > feature.persistentMute.enabled

Configure the Headset and Speakerphone

All VVX phones come with a handset and a dedicated connector for a headset and include support for a USB headset; all Polycom phones have built-in speakerphones. You can enable and disable each of these options, as shown in the following table. Note that although handsets are shipped with your phones, headsets are not provided.

VVX phones have a dedicated key to switch between speakerphone and headset. You can enable or disable the handsfree speakerphone mode.

> **Web Info: Configuring an external electronic hookswitch**
>
> You can configure all supported Polycom desktop phones with an external electronic hookswitch. For more information, see *Technical Bulletin 35150: Using an Electronic Hookswitch with SoundPoint IP and Polycom VVX 1500 Phones.*

Configure the Headset and Speakerphone

Parameter Function	template > parameter
Enable or disable headset memory mode.	**reg-advanced.cfg** and **site.cfg** > up.headsetMode

Configure the Headset and Speakerphone (continued)

Enable or disable handsfree speakerphone mode.	**reg-advanced.cfg** and **site.cfg** > up.handsfreeMode
Specify if the electronic hookswitch is enabled and what type of headset is attached.	**reg-advanced.cfg** and **site.cfg** > up.analogHeadsetOption
Specify if the handset or a headset should be used for audio.	**reg-advanced.cfg** and **site.cfg** > up.audioMode
Specify how phone and the USB headset interact.	**site.cfg** >up.headset.phoneVolumeControl
Specify if the USB headset volume persists between calls.	**site.cfg** > voice.volume.persist.headset

Example Handset, Headset, and Speakerphone Configuration

The following illustration shows the default settings in the reg-advanced.cfg template. In this example, handsfree mode is enabled and headset memory mode and electronic hookswitch are disabled.
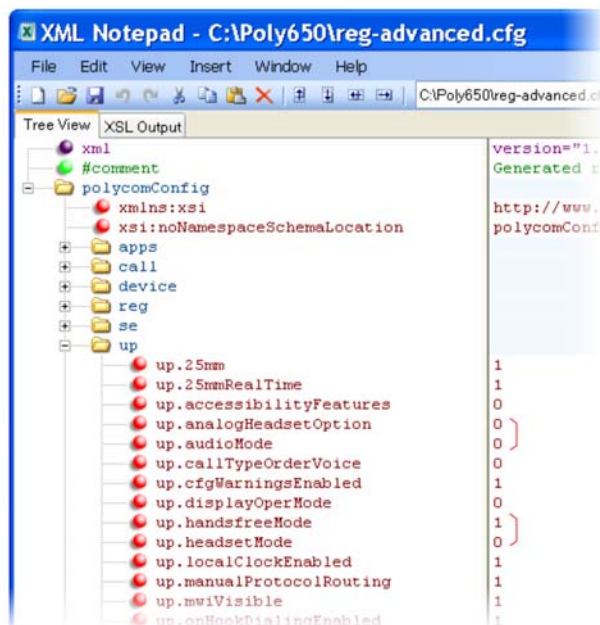
Apply Distinctive Ringing

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 following table shows you the parameters for each. If you set up distinctive ringing using more than one of the following methods, the phone uses the highest priority method.

● You can assign ringtones to specific contacts in the contact directory. For more information, see Apply Distinctive Incoming Call Treatment. This option is first and highest in priority.

● You can select a ringtone for each registered line on the phone. Press the **Menu** key, and select **Settings > Basic > Ring Type**. This option has the lowest 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.

> **Note:Using the SIP Alert-Info header to delay autoanswer**
>
> If you set **delay=0** in the **SIP.alert-Info** header, the phone immediately auto-answers incoming calls without ringing. If you set **delay=x** where x=time in seconds, the phone rings for that duration of time before auto-answering incoming calls.

Apply Distinctive Ringing

Parameter Function	template > parameter
Map the 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

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.

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 visually alerts you to an incoming call, but the phone does 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 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 displays on the phone and incoming calls continue to ring.

Configure Do Not Disturb

Parameter Function	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

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 so that when you turn on DND on the phone, incoming callers receive a busy signal.
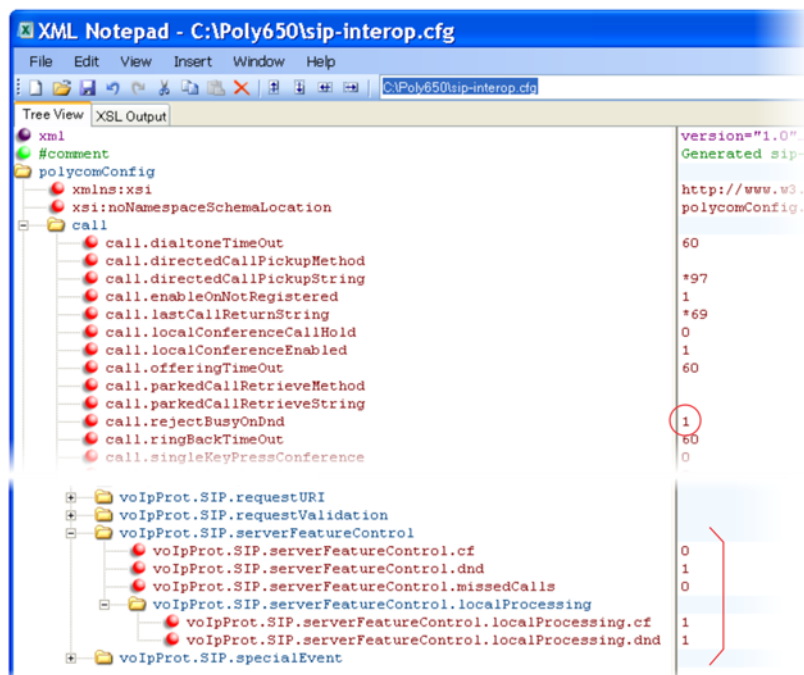
Note: DND LED alerts on the VVX

The LED on the Do Not Disturb key on the VVX 1500 is red when pressed or when server-based DND is enabled.

Configure Call Waiting Alerts

By default, the phone alerts 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

Parameter Function	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.
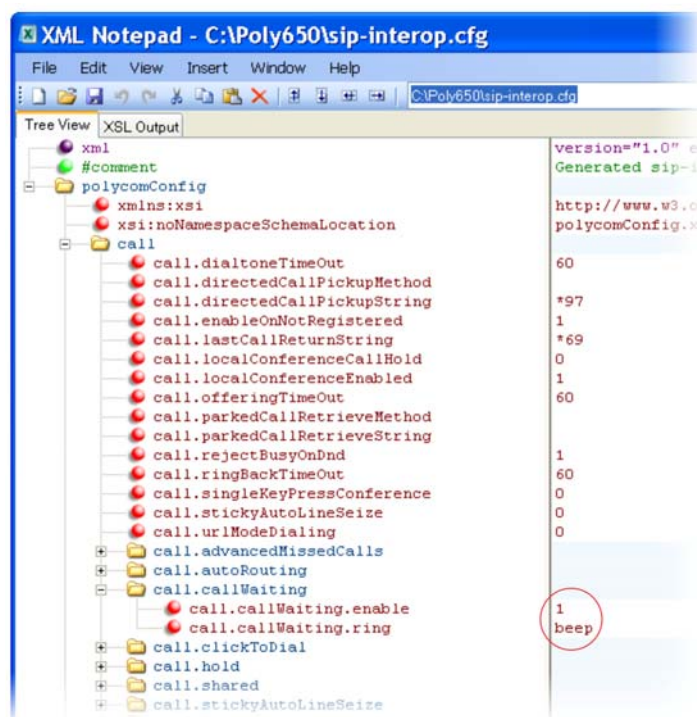
Configure Calling Party Identification

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. Note that the phone cannot match the identity of calling parties to entries in the corporate directory.

Configure Calling Party Identification

Parameter Function	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

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 displays on the phone screen.
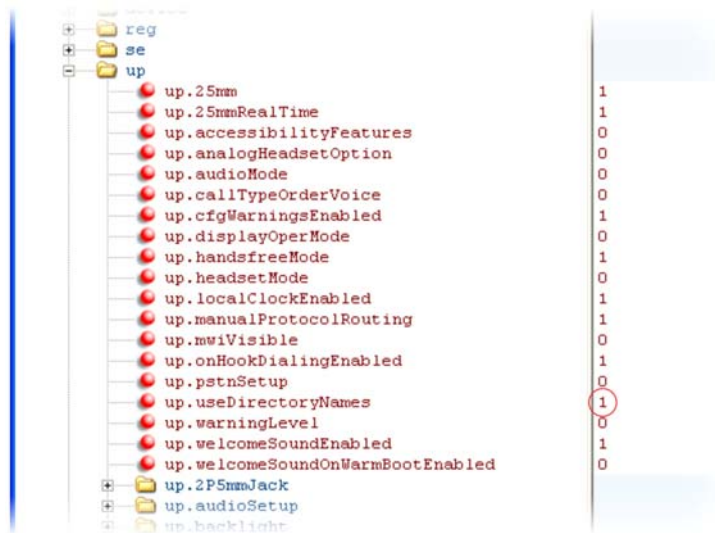
Enable Missed Call Notification

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 following table indicates, you can also configure the phone to record all missed calls or to display only missed calls that arrive through the SIP server. You can enable missed call notification for each registered line on a phone.

Enable Missed Call Notification

Parameter Function	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 display on line 1.
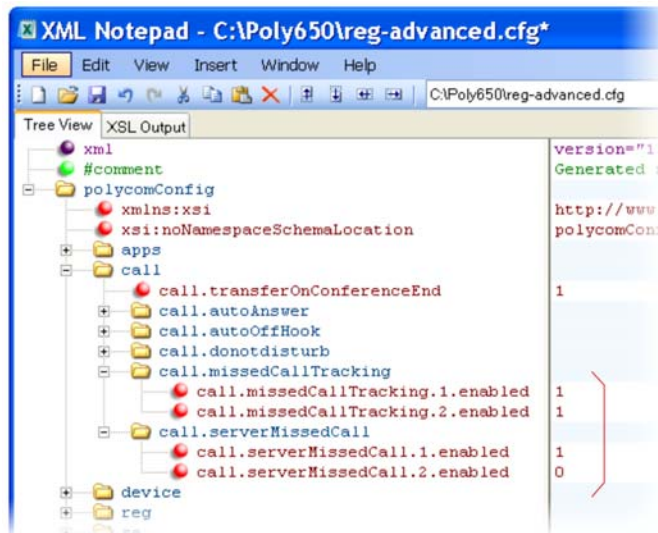
Apply Distinctive Incoming Call Treatment

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 are diverted to SIP address *3339951954* as specified in dc. Incoming calls from Bill Smith have been set to auto reject in ar and are sent to voicemail.
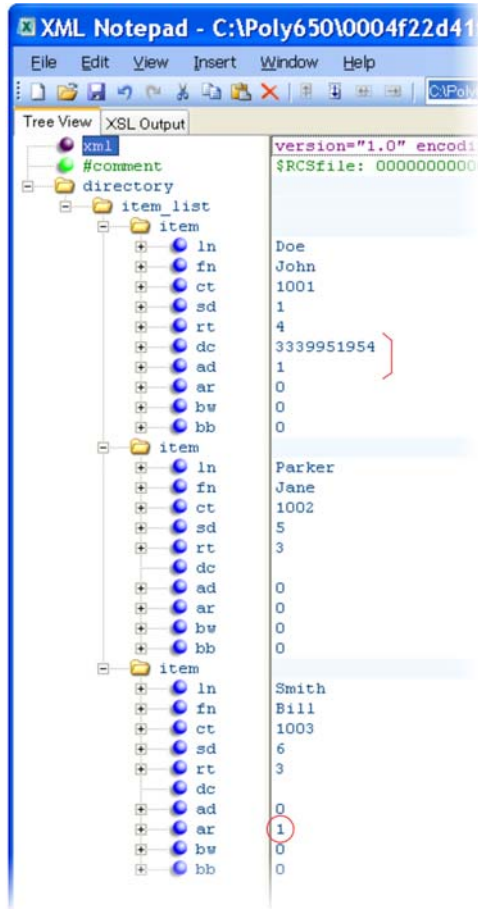
Note that if you enable both the auto divert and auto reject features, auto divert has precedence over auto reject. For a list of all parameters you can use in the contact directory, see the table Understanding the Local Contact Directory.

Apply Distinctive 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 following table shows you the parameters you can configure for this feature. This feature requires call server support.

Apply Distinctive Call Waiting

Parameter Function	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 cause the phone to ring in a manner specified by

`voIpProt.SIP.alertInfo.x.class`. In this example, the phone displays a visual LED notification, as specified by the value visual.

Synthesized Call Progress Tones

Polycom phones play 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 the Phone Display

This section provides information on setting up features available on the phone display screen.

Set Time Zone Location Description

The following two parameters configure a time zone location description for their associated GMT offset:

- `device.sntp.gmtOffsetcityID`

  If you are not provisioning phones manually from the phone menu or Web Configuration Utility and you are setting the `device.sntp.gmtOffset` parameter, then you must configure `device.sntp.gmtOffsetcityID` to ensure that the correct time zone location description displays on the phone menu and Web Configuration Utility. The time zone location description is set automatically if you set the `device.sntp.gmtOffset` parameter manually using the phone menu or Web Configuration Utility.

- `tcpIpApp.sntp.gmtOffsetcityID`

  If you are not provisioning phones manually from the Web Configuration Utility and you are setting the `tcpIpApp.sntp.gmtOffset` parameter, then you must configure `tcpIpApp.sntp.gmtOffsetcityID` to ensure that the correct time zone location description displays on the Web Configuration Utility. The time zone location description is set automatically if you set the `tcpIpApp.sntp.gmtOffset` parameter manually using the Web Configuration Utility.

Use the values in the following table to set the time zone location description.

Permitted Values		Default
0	(GMT -12:00) Eniwetok,Kwajalein	NULL
1	(GMT -11:00) Midway Island	
2	(GMT -10:00) Hawaii	
3	(GMT -9:00) Alaska	
4	(GMT -8:00) Pacific Time (US & Canada)	
5	(GMT -8:00) Baja California	
6	(GMT -7:00) Mountain Time (US & Canada)	
7	(GMT -7:00) Chihuahua,La Paz	
8	(GMT -7:00) Mazatlan	
9	(GMT -7:00) Arizona	
10	(GMT -6:00) Central Time (US & Canada)	
11	(GMT -6:00) Mexico City	
12	(GMT -6:00) Saskatchewan	
13	(GMT -6:00) Guadalajara	
14	(GMT -6:00) Monterrey	
15	(GMT -6:00) Central America	
16	(GMT -5:00) Eastern Time (US & Canada)	
17	(GMT -5:00) Indiana (East)	
18	(GMT -5:00) Bogota,Lima	
19	(GMT -5:00) Quito	
20	(GMT -4:30) Caracas	

Permitted Values		Default
21	(GMT -4:00) Atlantic Time (Canada)	
22	(GMT -4:00) San Juan	
23	(GMT -4:00) Manaus,La Paz	
24	(GMT -4:00) Asuncion,Cuiaba	
25	(GMT -4:00) Georgetown	
26	(GMT -3:30) Newfoundland	
27	(GMT -3:00) Brasilia	
28	(GMT -3:00) Buenos Aires	
29	(GMT -3:00) Greenland	
30	(GMT -3:00) Cayenne,Fortaleza	
31	(GMT -3:00) Montevideo	
32	(GMT -3:00) Salvador	
33	(GMT -3:00) Santiago	
34	(GMT -2:00) Mid-Atlantic	
35	(GMT -1:00) Azores	
36	(GMT -1:00) Cape Verde Islands	
37	(GMT 0:00) Western Europe Time	
38	(GMT 0:00) London,Lisbon	
39	(GMT 0:00) Casablanca	
40	(GMT 0:00) Dublin	
41	(GMT 0:00) Edinburgh	
42	(GMT 0:00) Monrovia	
43	(GMT 0:00) Reykjavik	
44	(GMT +1:00) Belgrade	
45	(GMT +1:00) Bratislava	
46	(GMT +1:00) Budapest	
47	(GMT +1:00) Ljubljana	
48	(GMT +1:00) Prague	
49	(GMT +1:00) Sarajevo,Skopje	
50	(GMT +1:00) Warsaw,Zagreb	
51	(GMT +1:00) Brussels	
52	(GMT +1:00) Copenhagen	
53	(GMT +1:00) Madrid,Paris	
54	(GMT +1:00) Amsterdam,Berlin	
55	(GMT +1:00) Bern,Rome	
56	(GMT +1:00) Stockholm,Vienna	
57	(GMT +1:00) West Central Africa	
58	(GMT +1:00) Windhoek	
59	(GMT +2:00) Bucharest,Cairo	
60	(GMT +2:00) Amman,Beirut	

Permitted Values		Default
61	(GMT +2:00) Helsinki,Kyiv	
62	(GMT +2:00) Riga,Sofia	
63	(GMT +2:00) Tallinn,Vilnius	
64	(GMT +2:00) Athens,Istanbul	
65	(GMT +2:00) Damascus	
66	(GMT +2:00) E.Europe	
67	(GMT +2:00) Harare,Pretoria	
68	(GMT +2:00) Jerusalem	
69	(GMT +2:00) Kaliningrad (RTZ 1)	
70	(GMT +2:00) Tripoli	
71	(GMT +3:00) Moscow	
72	(GMT +3:00) St.Petersburg	
73	(GMT +3:00) Volgograd (RTZ 2)	
74	(GMT +3:00) Kuwait,Riyadh	
75	(GMT +3:00) Nairobi	
78	(GMT +3:00) Baghdad	
76	(GMT +3:00) Minsk	
77	(GMT +3:30) Tehran	
79	(GMT +4:00) Abu Dhabi,Muscat	
80	(GMT +4:00) Baku,Tbilisi	
81	(GMT +4:00) Izhevsk,Samara (RTZ 3)	
82	(GMT +4:00) Port Louis	
83	(GMT +4:00) Yerevan	
84	(GMT +4:30) Kabul	
85	(GMT +5:00) Ekaterinburg (RTZ 4)	
86	(GMT +5:00) Islamabad	
87	(GMT +5:00) Karachi	
88	(GMT +5:00) Tashkent	
89	(GMT +5:30) Mumbai,Chennai	
90	(GMT +5:30) Kolkata,New Delhi	
91	(GMT +5:30) Sri Jayawardenepura	
92	(GMT +5:45) Kathmandu	
93	(GMT +6:00) Astana,Dhaka	
94	(GMT +6:00) Almaty	
95	(GMT +6:00) Novosibirsk (RTZ 5)	
96	(GMT +6:30) Yangon (Rangoon)	
97	(GMT +7:00) Bangkok,Hanoi	
98	(GMT +7:00) Jakarta	
99	(GMT +7:00) Krasnoyarsk (RTZ 6)	
100	(GMT +8:00) Beijing,Chongqing	

Permitted Values		Default
101	(GMT +8:00) Hong Kong,Urumqi	
102	(GMT +8:00) Kuala Lumpur	
103	(GMT +8:00) Singapore	
104	(GMT +8:00) Taipei,Perth	
105	(GMT +8:00) Irkutsk (RTZ 7)	
106	(GMT +8:00) Ulaanbaatar	
107	(GMT +9:00) Tokyo,Seoul,Osaka	
108	(GMT +9:00) Sapporo,Yakutsk (RTZ 8)	
109	(GMT +9:30) Adelaide,Darwin	
110	(GMT +10:00) Canberra	
111	(GMT +10:00) Magadan (RTZ 9)	
112	(GMT +10:00) Melbourne	
113	(GMT +10:00) Sydney,Brisbane	
114	(GMT +10:00) Hobart	
115	(GMT +10:00) Vladivostok	
116	(GMT +10:00) Guam,Port Moresby	
117	(GMT +11:00) Solomon Islands	
118	(GMT +11:00) New Caledonia	
119	(GMT +11:00) Chokurdakh (RTZ 10)	
120	(GMT +12:00) Fiji Islands	
121	(GMT +12:00) Auckland,Anadyr	
122	(GMT +12:00) Petropavlovsk-Kamchatsky (RTZ 11)	
123	(GMT +12:00) Wellington	
124	(GMT +12:00) Marshall Islands	
125	(GMT +13:00) Nuku'alofa	
126	(GMT +13:00) Samoa	

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 have to synchronize the phone to the Simple Network Time Protocol (SNTP) time server. Until a successful SNTP response is received, the phone continuously flashes the time and date to indicate that they are not accurate.

The time and date display on phones in PSTN mode are 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.

See the following table for basic time and display parameters.

Set the Time and Date Display

Parameter Function	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

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.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

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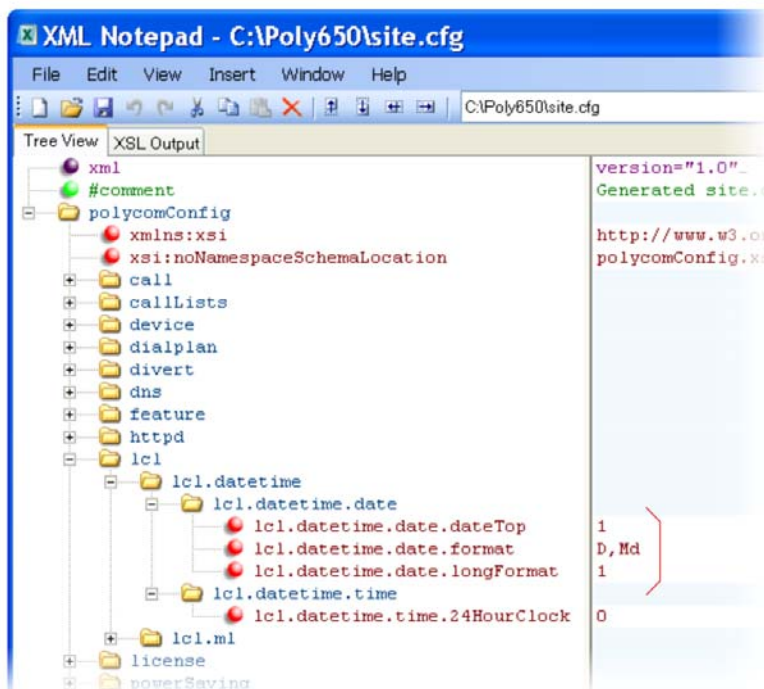
Use the table Date Formats to choose values for the `lcl.datetime.date.format` and `lcl.datetime.date.longformat` parameters. The table shows values for Friday, August 19, 2011.

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

Date Formats

Md,D	1	August 19, Friday
D,dM	0	Fri, 19 Aug
D,dM	1	Friday, August 19
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

Configure a Phone Theme

You can configure a Classic or Modern theme for the interface of the VVX 500 and 600 business media phone.

Phone Theme Parameters

The following table lists parameters that configure a theme for the VVX 500 and 600 business media phones.

Phone Theme Parameters

Parameter Template	Permitted Values
`device.theme` device.cfg	Classic (default) - The phone uses the Classic theme. Modern - The phone uses the Modern theme.
`device.theme.set` device.cfg	1 (Default) - The phone supports both the Classic and Modern theme. The device.theme parameter specifies which theme to use. 0 - The phone supports only Modern theme.

Configure Off-Hook Call Status

On the VVX 500, 501, 600, and 601 business media phones, you can configure the default phone screen as follows:

- You can configure the phone to display the active call screen or line view when in an active call. Configuring display of the line view enables users to see the status of any lines, buddies, and BLF contacts they are monitoring.

- When the phone is off-hook, you can configure the phone to display the Home screen or line view. Configuring display of the line view when the phone goes off hook enables users to quickly select a contact to dial and includes favorites and BLF lines. In this scenario, when users start to enter a number from the keypad, the phone switches to the dialer screen.

Default Phone Screen Parameters

Parameter Template	Permitted Values
`up.LineViewCallStatus.enabled` features.cfg	0 (Default) - In an active call, the active call screen displays. Any incoming or outgoing call triggers display of the active call screen. 1 - During an incoming call and in an active call, the line view displays and call details display on the status ribbon.
`up.LineViewCallStatusTimeout` features.cfg	Specify the timeout period after which the phones go back to the Line Screen when the user goes to the Active Call Screen from the Line View. Available on VVX 500 and 600 business media phones. 10 (default) - The phone returns to the line screen after 10 seconds. 2 - 10 - Specify the seconds after which the phone returns to the line screen.
`up.OffHookLineView.enabled` features.cfg	0 (Default) - After the phone goes off-hook, the phone displays the dialing screen. 1 - After the phone goes off hook, the phone displays the line screen.

Set a Graphic Display Background

You can display a graphic image on the background of the all VVX business media phones and connected VVX Color Expansion Modules. The table Set a Graphic Display Background lists parameters you must configure to set the graphic display background on VVX business media phones and connected expansion modules. Note that the background image you configure displays across the entire phone screen; the time and date, and line key and soft key labels display over the background. If you want the background image to display more visibly from behind line key labels, use `up.transparentLines` to render line key labels transparent - this option is available only on the VVX 500/501 and 600/601 business media phones.

For VVX phones:

- The VVX phones display a default background picture. You can select your own background picture or design, or you can import a custom image. You can also select images from a USB using the picture frame feature as shown in the section Configure the Digital Picture Frame.

- All Polycom devices support JPEG, BMP, and PNG image file formats; progressive/multiscan JPEG images are not supported.

- The maximum image size varies with the phone LCD screen size as listed in the following table.

Phone Screen Sizes

Phone	Screen Size
VVX 300 series	208x104 pixels (Grayscale)
VVX 400 series	320x240 pixels
VVX 500 series	320x240 pixels
VVX 600 series	480x272 pixels

Phone Screen Sizes

VVX 1500	800x480 pixels
VVX Color Expansion Module	272x480 pixels

Web Info: Adding a graphic display background

For detailed instructions on adding a graphic display to a VVX phone, see the *Polycom VVX Business Media Phones User Guide*.

Set a Graphic Display Background

Parameter Function	template > parameter
Specify a background to display for your phone type.	**features.cfg** > bg.*
Enable or disable transparent line key labels on the VVX 500/501 and 600/601.	**features.cfg** > up.transparentLines

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Example Graphic Display Background Configuration

This example configuration shows a background image applied to a VVX phone and expansion module. The default background in the **features.cfg** template file, specified in the bg.color.selection parameter, is set to 2,1, where 2 enables background images and 1 selects the image. For example, 1= bg.color.bm.1.em.name and bg.color.bm.1.name. The phone displays the background image, in this case lighthouse.jpg, and the expansion module displays the tulips.jpg.
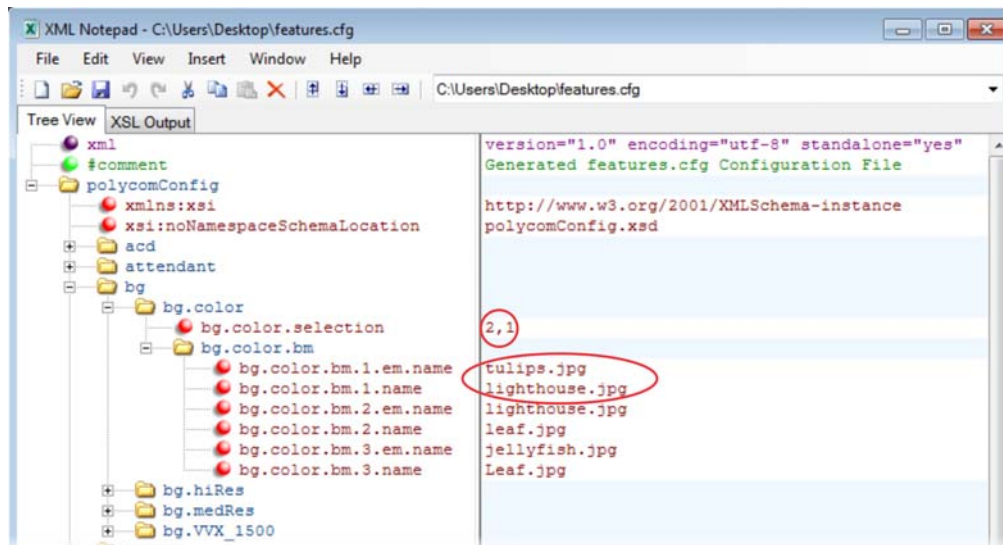
This example configuration results in the following graphic display background on the phone and expansion module. Note that line and soft key labels display over the background image.

Enable Background Image Lock

Administrators can disable the user option, available in the digital picture frame feature, to set images as a background when viewing images on a USB attached to the phone. By default, users can set the background image. Disabling this feature removes the following options when logged into the phone as a user:

● On the phone, the Background menu at **Home > Settings > Basic > Preferences > Background**.

● On the phone, the icon to set image as background at S**ettings > Features > Removable Storage Media > Picture Frame**. On VVX 1500 phones, at **Menu > Features > Removable Storage**.

● On the Web Configuration Utility, the Background option in the Preferences menu.

Background Image Lock Parameters

Parameter Function	template > parameter
Enable or disable the option to set an image as background.	**features.cfg** > bg.background.enabled

Configure the Digital Picture Frame

On the VVX phones, you can display a slide show of images on the phone's idle screen. Images must be saved in JPEG, BMP, or PNG format on a directory on a USB device that is attached to the phone. The parameters you can configure are listed in the table Configure the Digital Picture Frame. The phone can display a maximum image size of 9999x9999 pixels and a maximum of 1000 images.

> **Note: Maximum image size**
> Although 9999x9999 images and progressive/multiscan JPEG images are supported, the maximum image size that can be downloaded is restricted by the available memory in the phone.

Configure the Digital Picture Frame

Parameter Function	template > parameter
To enable or disable the digital picture frame.	**features.cfg** > feature.pictureFrame.enabled
Specify the name of the folder on the USB device containing the images.	**reg-advanced.cfg** > up.pictureFrame.folder
Set how long each picture displays.	**reg-advanced.cfg** > up.pictureFrame.timePerImage

Example Digital Picture Frame Configuration

In the following illustration, the digital picture frame feature is enabled in the **features.cfg** template file.

In the **reg-advanced.cfg** template file, the phone looks on the USB device for images in the folder named pictures and each picture displays for seven seconds.
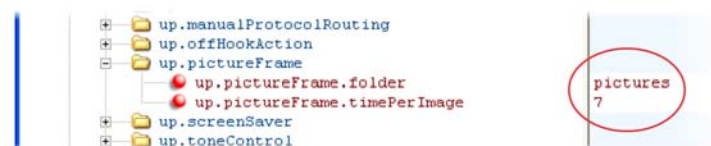
After the configuration is complete, restart the phone, insert the USB device to the phone. A Removable Storage Media icon displays on the phone's screen.

To show your pictures, press the icon and press **Picture Frame**.

> **Note: Accessing the digital picture frame**
> You can access the digital picture frame using *PicFrame:// URL*.

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 **VVXLocalization** folder. This folder is included with the Polycom UC Software you downloaded to your provisioning server. If you want to edit the language files, you must 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/>.

All phones support the following the following languages: Arabic, Simplified Chinese, Traditional Chinese, Danish, Dutch, English, French, German, Italian, Japanese, Korean, Norwegian, Polish, Brazilian Portuguese, Russian, Slovenian, International Spanish, and Swedish.

> **Note: Multilingual support for the updater**
> At this time, the updater is available in English only.

Set the Phone Language

Parameter Function	**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

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

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.
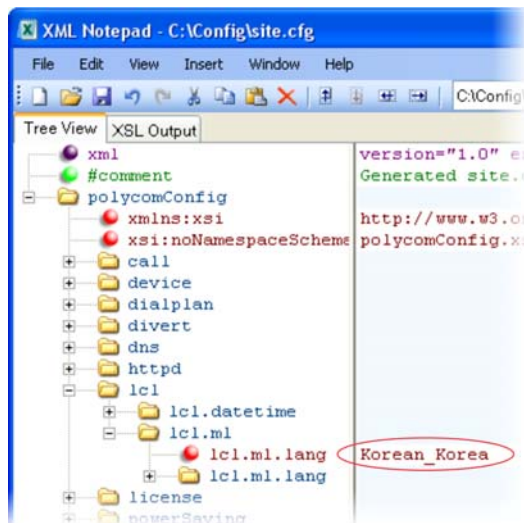
After configuration, the phone uses Korean characters.

Use Pinyin Text Input

Pinyin is the phonetic system used to transcribe Mandarin pronunciation of Chinese into Latin characters. The pinyin text input feature on Polycom VVX business media phones uses Nuance XT9® Smart Input to enable you to enter Chinese characters into text input fields using the phone's dial pad keys or through the onscreen keyboard.

The pinyin text input feature is available on the following Polycom phones:

● VVX 1500 running Polycom UC Software 4.0.2B or later
● VVX 300 series, 400 series, 500 series, and 600 series phones running UC Software 5.0.0 or later

You can enter pinyin text using the pinyin text input widget or the onscreen keyboard. To use the pinyin text input feature on Polycom phones, administrators must download a license key to the user's phone.

Web Info: Pinyin text input

For complete information on the pinyin text input feature, see the *Polycom VVX Business Media Phones User Guide*.

Unique Line Label for Registration Lines

Administrators can configure line registrations to display for multiple line keys. When using this feature with the parameter reg.x.label.y and x=2 or higher, multiple line keys display for the registered line address. If you configure the line to display on multiple line keys without a unique label assigned to each line, the lines are labeled automatically in numeric order. For example, if you have four line keys for line 4144 labeled Polycom, the line keys are labeled as 1_Polycom, 2_ Polycom, 3_ Polycom, and 4_ Polycom. This also applies to lines without labels.

Configure Unique Line Labels

Parameter Function	template > parameter
Configure a unique line label for multiple line keys.	**reg-advanced.cfg, site.cfg** > reg.x.line.y.label
Determines the label that displays on the line key.	**features.cfg** > up.cfgLabelElide
Determines the label that displays on the line key.	**features.cfg** > up.cfgUniqueLineLabel

Set Patterns for LED Indicators

The LED indicators on Polycom devices and expansion modules alert users to the different phone states and status of remote contacts. You can turn LED indicators on or off, and set the pattern, color, and duration of a pattern for all physical keys on the phones.

You can set the pattern, color, and duration for the following LED indicators:

- Line keys
- Message Waiting Indicator (MWI)
- Headset key (excluding VVX 101 and 201)

The following tables list the parameters that configure the pattern state, color, and duration of the LED indicators and pattern types on Polycom devices and expansion modules. Configure these parameters as follows:

- For x, specify a pattern type listed in the table LED Indicator Pattern Type.
- For y, specify a pattern number between 1-20.

Configure the LED Indicator Patterns

Parameter Function	template > parameter
Turns the LED indicator on or off depending on the pattern's state.	**features.cfg** > ind.pattern.x.step.y.state
Indicates the color of the line key LED indicators.	**features.cfg** > ind.pattern.x.step.y.color
Sets the duration of the pattern of the LED indicator.	**features.cfg** > ind.pattern.x.step.y.duration

LED Indicator Pattern Type

Pattern Type	Function
powerSaving	Sets the behavior for Message Waiting Indicator when the phone is in Power Saving mode.
active	Sets the pattern for line keys during active calls.
on	Turns on the LED indicator pattern.
off	Turns off the LED indicator pattern.
offering	Sets the pattern for line keys during incoming calls.
flash	Sets the pattern for line keys during held calls and the Message Waiting Indicator when there are unread voicemail messages.
lockedOut	Sets the pattern for line keys when a remote party is busy on a shared line.
FlashSlow	Sets the pattern for the Headset key when Headset Memory Mode is enabled.
held	Sets the pattern for line keys during a held call.
remoteBusyOffering	Sets the pattern for line keys for monitored BLF contacts.

LED Pattern Examples

This section includes example configurations you can use to set the patterns of LED indicators for Polycom devices and expansion modules.

Disable the Headset Key LED in Headset Memory Mode

By default, the Headset key on all VVX phones, excluding VVX 101 and 201, glows green for analog headsets and blue for USB headsets. The Headset key also flashes by default if Headset Memory Mode is enabled. You can disable and turn off the flash pattern for the Headset key when Headset Memory Mode is enabled.

By default, the following parameters set the behavior of the MWI during Power Saving mode:

- `ind.pattern.flashSlow.step.1.state` = 1—turns on the LED indicator.
- `ind.pattern.flashSlow.step.1.duration` = 100—sets the duration of the pattern.
- `ind.pattern.flashSlow.step.2.state` = 0—turns off the LED indicator for the second step.
- `ind.pattern.flashSlow.step.2.duration` = 2900 —sets the duration for how long the LED indicator is off before the pattern repeats.

To disable the flash pattern for the Headset key:

» Set the parameter `ind.pattern.flashSlow.step.1.state` to 0.

Configure the Device LED Pattern

The following example configures the LED 'active' pattern to alternate between green and red once per second.

To configure the device LED to the 'active' pattern:

```
ind.pattern.active.step.1.color="Green"
ind.pattern.active.step.1.state="On"
ind.pattern.active.step.1.duration=1000
ind.pattern.active.step.2.color="Red"
ind.pattern.active.step.2.state="On"
ind.pattern.active.step.2.duration=1000
ind.pattern.active.step.3.color="Green"
ind.pattern.active.step.3.state="On"
ind.pattern.active.step.3.duration=1000
```

Turn Off the Message Waiting Indicator in Power Saving Mode

When Power Saving mode is enabled, the screen darkens, and the MWI flashes red. By default, the powerSaving pattern has two steps before the pattern is repeated: a quick on period and then a long off period. You can turn off the MWI or change the duration of the pattern steps.

By default, the following parameters set the behavior of the MWI during Power Saving mode:

- `ind.pattern.powerSaving.step.1.state` = 1 —turns on the LED indicator.

● `ind.pattern.powerSaving.step.1.duration` = 100—sets the duration of the pattern.

● `ind.pattern.powerSaving.step.2.state` = 0— turns off the LED indicator for the second step.

● `ind.pattern.powerSaving.step.2.duration` = 2900 —sets the duration for how long the LED indicator is off before the pattern repeats.

To disable the pattern for the MWI in Power Saving mode:

» Set the parameter `ind.pattern.powerSaving.step.1.state` to 0.

Change the Color of Line Key Indicators for Incoming Calls

When a phone receives an incoming call, the line key LED indicator flashes green. You can change the color of the indicator to Yellow or Red for incoming calls.

By default, the following parameters set the behavior of the line key LED indicators for incoming calls:

● `ind.pattern.offering.step.1.state` = 1—turns on the LED indicator.

● `ind.pattern.offering.step.1.duration` = 250—sets the duration of the pattern.

● `ind.pattern.offering.step.1.color` = Green—sets the color of the LED indicator for the pattern.

● `ind.pattern.offering.step.2.state`= 0— turns off the LED indicator for the second step

● `ind.pattern.offering.step.2.duration`= 250—sets the duration for how long the LED indicator is off before the pattern repeats.

To change the color of the line key indicator:

» Set the parameter `ind.pattern.offering.step.1.color` to Yellow.

The following figure shows a Yellow line key LED indicator for an incoming call on a VVX 500/501 phone.
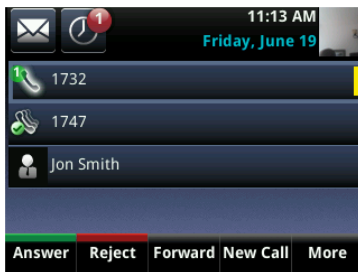
Configure Call Controls

This section provides information on configuring general phone call controls.

Enable Last Call Return

The phone supports redialing of the last received call. The following table 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

Parameter Function	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:
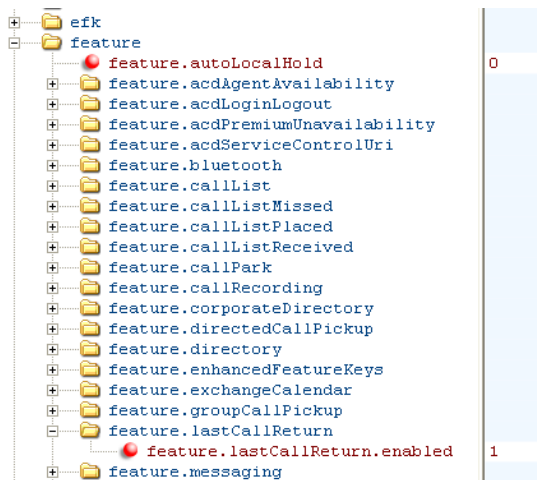
When last call return is enabled, you can configure the feature using parameters located in the **sip-interop.cfg** template file. The following shows the default value for the `call.lastCallReturnString` parameter. The last call return string value depends on the call server you use. Consult with your call server provider for the last call return string.
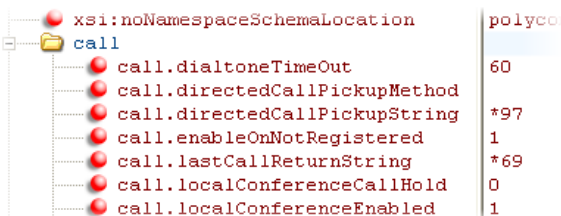
When you enable the last call return feature, the phone displays an **LCR** soft key when it goes off-hook. When you press the **LCR** soft key, you place a call to the phone address that last called you.

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

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 following

table for a list of available parameters you can configure for this feature. When you place an active call on hold, a message informs 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 RFC Music on Hold draft-worley-service-example.

Enable Call Hold

Parameter Function	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

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

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 begins to play 45 seconds after you put a party on hold, and that the tone repeats every 30 seconds.
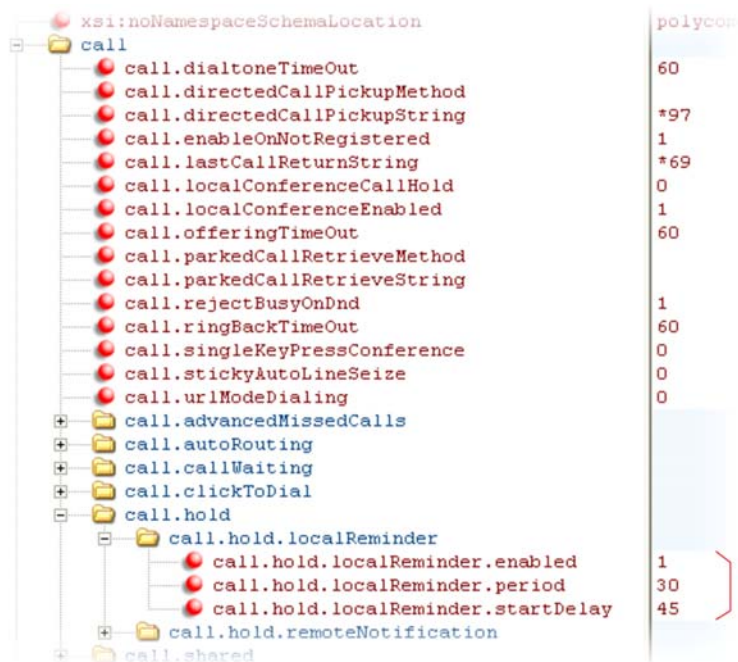
In the second illustration, the `musicOnHold.uri` parameter has been configured so the party on hold hears music played from SIP URI *moh@example.com*.
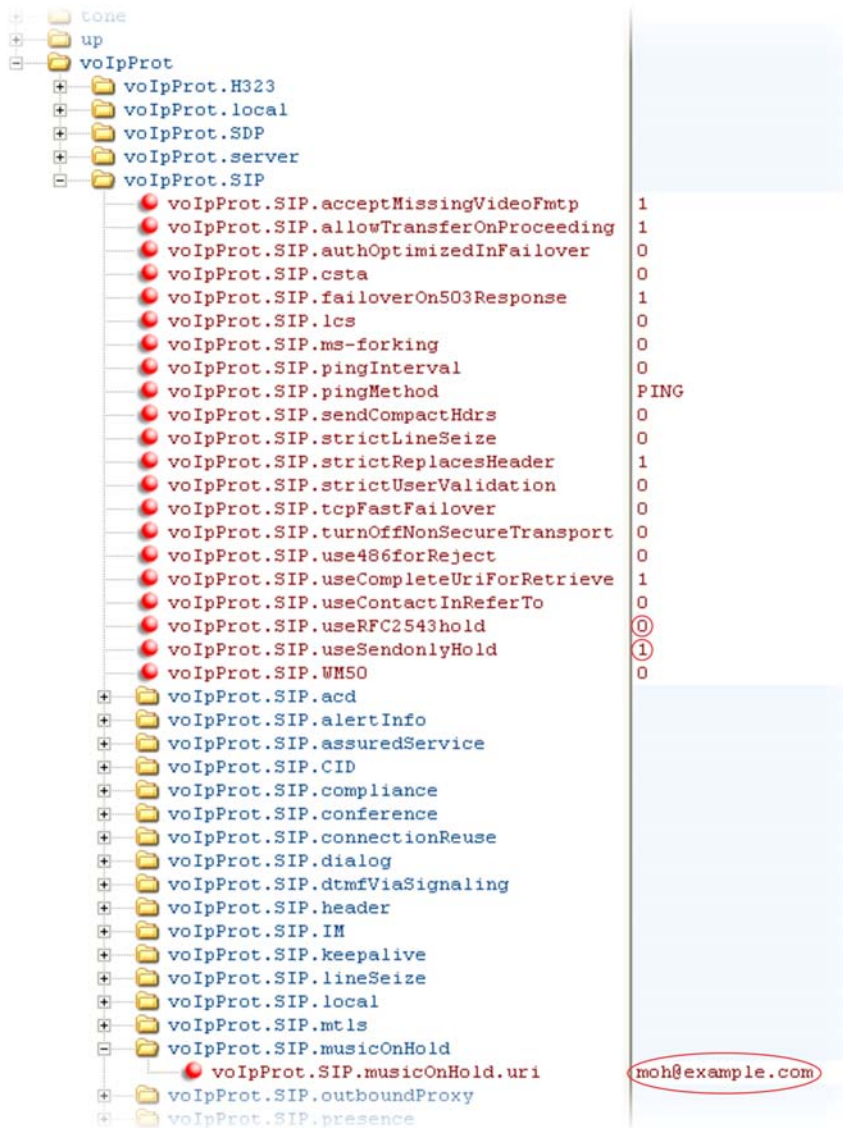
Configure Call Park and Retrieve

You can park an active call on a separate call orbit and retrieve parked calls from the call orbit on 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

Parameter Function	template > parameter
Enable or disable call park and retrieve.	**features.cfg** > feature.callPark.enabled
Specify the method the phone uses to retrieve a BLF call.	**sip-interop.cfg** > call.parkedCallRetrieveMethod
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.
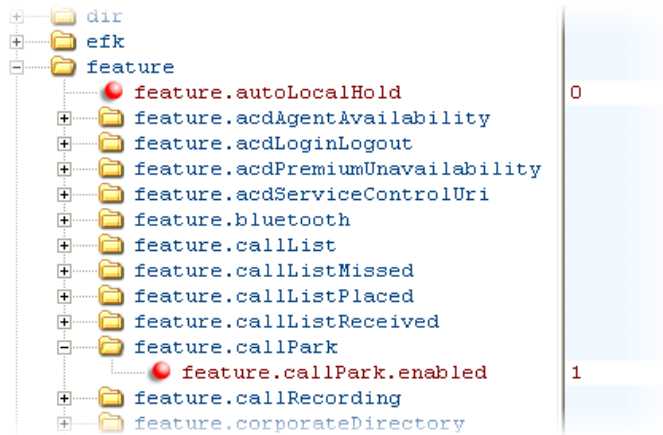
You can configure the call park and call retrieve feature using parameters located in the **sip-interop.cfg** template file. The following illustration shows that the parked call retrieve method has been set to native, meaning that the phone uses SIP INVITE with the Replaces header. The method can also be set to legacy, meaning that the phone uses the call.parkedCallRetrieveString star code to retrieve the parked call.
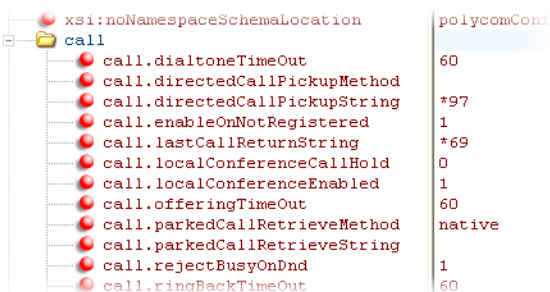
When the call park and retrieve feature is enabled, the **Park** soft key displays when you are in a connected call. To park the call, press the **Park** soft key and enter the number of the call orbit and park the call.

To retrieve a parked call, go off-hook and press the **Pickup** soft key. Enter the number of the call orbit and press the **More** and **Retrieve** soft keys, shown next.
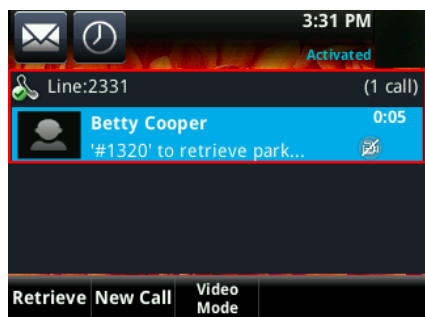
Use Call Transfer

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 begin a new call and party A disconnects. You can also set the default transfer type. The following table 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 displays 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

Parameter Function	template > parameter
Specify whether to allow transfers while calls are in a proceeding state.	**sip-interop.cfg** > voIpProt.SIP.allowTransferOnProceeding
Set the default transfer type the phone uses when transferring a call.	**features.cfg** > call.DefaultTransferType

Example Call Transfer Configuration

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

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 can 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 are not affected.

Server-based call forwarding behaves the same as pre-SIP 2.1 feature with the following exception:

● If server-based call forwarding is enabled, but inactive, and you press the Forward soft key, the 'moving arrow' icon does not display on your phone and incoming calls are not forwarded.

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, you must 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.

If you are registering your Polycom phones with Skype for Business Server, the following types of call forwarding are available on Skype for Business-enabled Polycom phones:

● Disable Call Forwarding

- Forward to a contact
- Forward to voicemail

Configure Call Forwarding

Parameter Function	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.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

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 is used if the parameters `divert.busy`, `divert.dnd`, or `divert.noanswer` are not set. Parameters you set in those fields override `divert.1.contact`.

To enable these three divert options for each registration, you must 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 are 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`.
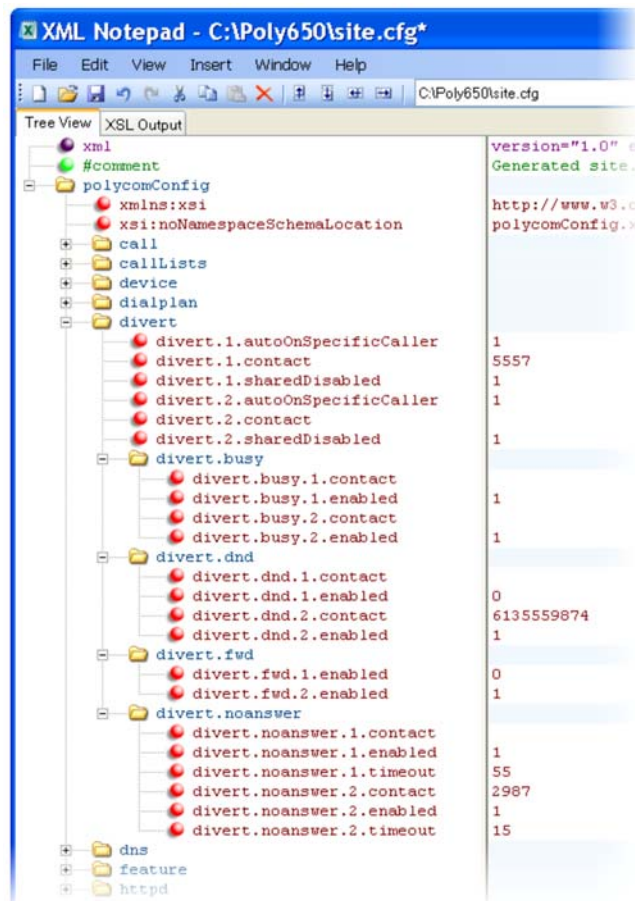
Enable Automatic Off-Hook Call Placement

You can configure the phone 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 lift the handset, press the New Call soft key, or press the headset or speakerphone buttons on the phone. As shown in the following table, you can specify an off-hook call contact and enable or disable the feature for each registration.

If you are using the VVX 500 series, 600 series, or 1500 phones, you can specify whether the automatic call uses the SIP (audio only) protocol or the H.323 (video) protocol.

Enable Automatic Off-Hook Call Placement

Parameter Function	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
Specify the call protocol to use for the VVX 500/501, 600/601, and 1500.	**reg-advanced** > call.autoOffHook.x.protocol

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 automatically placed 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 automatically placed to *6417*. On VVX 500 series, 600 series, and 1500 phones, registration 2 automatically places a call using the H.323 protocol instead of the SIP protocol. Other phones ignore the protocol parameter.

```
    xsi:noNamespaceSchemaLocation          polycomConfig.xsd
  apps
  call
        call.transferOnConferenceEnd      1
      call.autoAnswer
      call.autoOffHook
            call.autoOffHook.1.contact    6416@polycom.com
            call.autoOffHook.1.enabled    1
            call.autoOffHook.1.protocol
            call.autoOffHook.2.contact    6417
            call.autoOffHook.2.enabled    1
            call.autoOffHook.2.protocol   H.323
      call.donotdisturb
      call.missedCallTracking
```
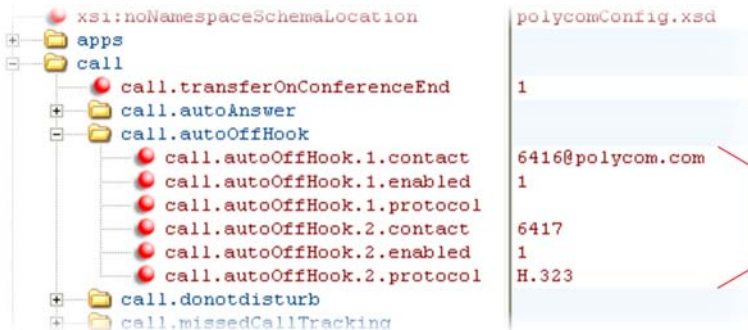
Configure Directed Call Pickup

Directed call pickup 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.

To enable or disable this feature for Sylantro call servers, set:

- `feature.directedCallPickup.enabled=1`

To configure this feature for all other call servers, use the parameters:

- `call.directedCallPickupMethod`
- `call.directedCallPickupString`

The following table lists the configuration parameters for the directed call pick-up feature.

Configure Directed Call Pickup

Parameter Function	template > parameter
Enable or disable Directed Call Pickup for Sylantro call Servers.	**features.cfg** > feature.directedCallPickup.enabled
Specify the type of directed call pick-up.	**sip-interop.cfg** > call.directedCallPickupMethod
Specify the star code to initiate a directed call pickup.	**sip-interop.cfg** > call.directedCallPickupString
Determine the type of SIP header to include.	**sip-interop.cfg** > voIpProt.SIP.strictReplacesHeader

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:
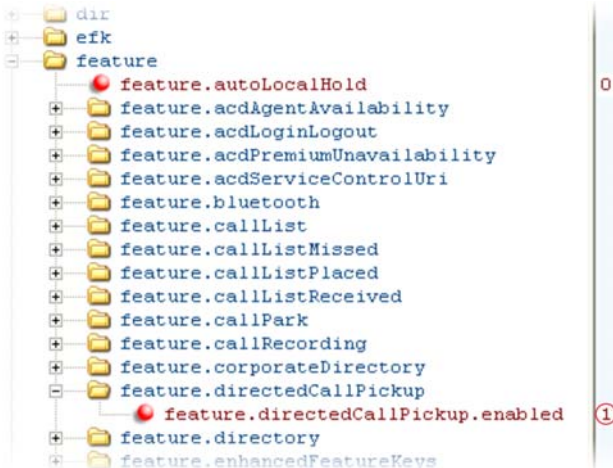
Once directed call pickup is enabled, you can configure the feature using parameters located in the **sip-interop.cfg** template file. In the following illustration, the pickup method has been set to `native`, which means that the server is used for directed call pickup instead of the `PickupString`. 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.
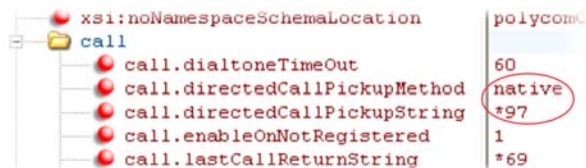
When you enable directed call pickup, the phone displays a **Pickup** soft key when you go off-hook. When you press the **Pickup** soft key, the **Directd** soft key displays.

Enable Multiple Registrations

Polycom phones can have multiple registrations; each registration requires an address, or phone number. Polycom phones registered with Microsoft Skype for Business Server support one Skype for Business registration. The maximum number of registrations vary by phone and are listed in the following table. The maximum registrations listed are supported with UC Software 4.0.1 and later.

Maximum Number of Registrations Per Phone

Phone Model Name	Maximum Registrations
VVX 101	One (1)

Maximum Number of Registrations Per Phone

VVX 201	Two (2)
VVX 300/301/310/311	Six (6)
VVX 400/401/410/411	Eight (8)
VVX 500/501	Twelve (12)
VVX 600/601	Sixteen (16)
VVX 1500	Twenty four (24)

You can also add up to three VVX Expansion Modules to a single VVX 300 series, 400 series, 500 series, or 600 series phone to increase the total number of registrations to 34. For more information, see Connect Polycom VVX Expansion Modules.

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 following table.

Enable Multiple Registrations

Parameter Function	**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 are used instead. The optional array of servers and their parameters override the servers specified in <voIpProt.server/> if non-Null.	**reg-basic.cfg**, **reg-advanced.cfg** > reg.x.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

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.
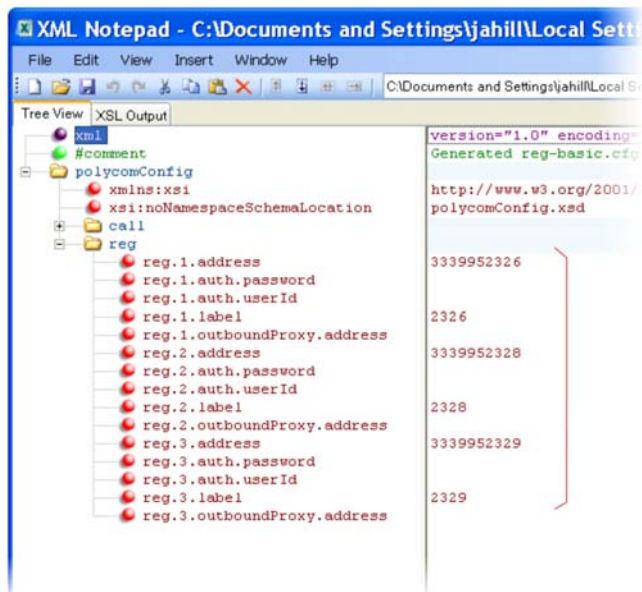
In the **reg-advanced.cfg** template shown next, when you make a call using line 1, the name you enter in `reg.1.displayname` displays as your caller ID, in this case *Lisa*. The parameter `reg.x.type` is left in the default `private`, which indicates that the registration uses standard call signaling.
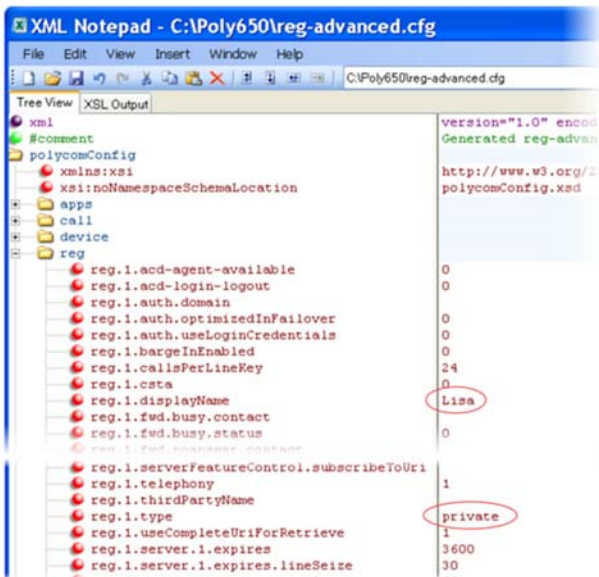
This configuration results in the following registrations on a VVX 600/601 phone:

Assign Multiple Line Keys Per Registration

You can assign a single registered phone line address to multiple line keys on VVX business media phones. This feature is not supported when registered with Microsoft Skype for Business Server.

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. For the maximum number of line keys per registration for each phone model, and for definitions of all features associated with flexible call appearances, refer to the following table.

Multiple Line Keys Per Registration

Parameter Function	template > parameter
Specify the number of line keys to use for a single registration.	**reg-advanced.cfg** > reg.x.lineKeys

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 *2346*.

The phone displays the registered line address *2346* on four line keys, as shown next.
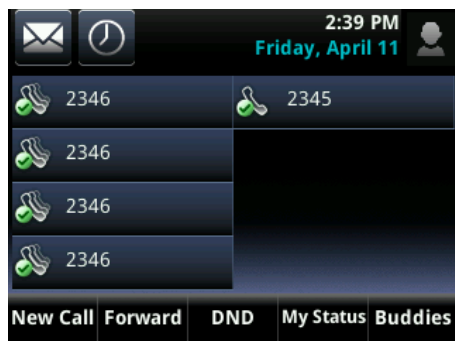
Configure Shared Call Appearances

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

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

> **Tip: Shared call and bridged line appearance 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

Parameter Function	**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 subscribes to a server providing call state information.	**reg-advanced.cfg** > reg.x.*

Configure Shared Call Appearances

Specify per-registration whether diversion should be disabled on shared lines.   **sip-interop.cfg** > divert.x.sharedDisabled

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

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.
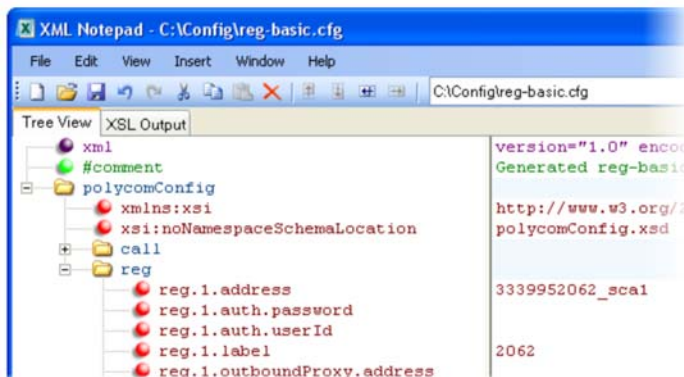
If you want to configure this line to be shared, in the **reg-advanced.cfg** template, specify shared in `reg.1.type`. All phones that specify shared for registration 1 have shared call appearance enabled for this line. In the following example, the `reg.1.bargeInEnabled` parameter is set to '1' to enable phones of this group to barge in on active calls.
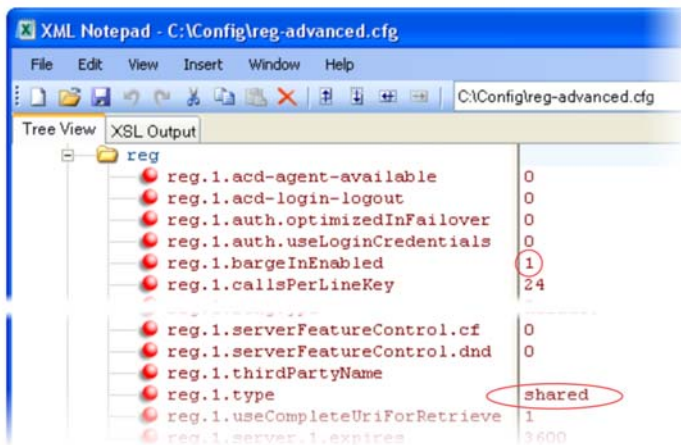
After setting these parameters, activity on line *2062* displays on all phones that configure a shared call appearance for line *2062*, as shown in the following illustration.
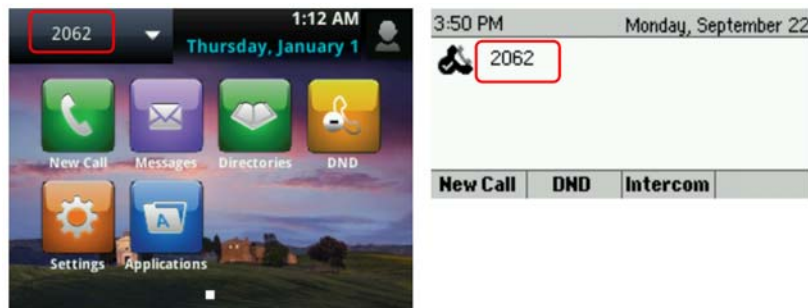
Call Forward on Shared Lines

You can enable server-based call forwarding on shared lines on the RealPresence Trio 8800 system and VVX business media phones. If using BroadWorks R20 server, note the following

- Local call-forwarding is not supported on shared lines on the BroadWorks R20 server.
- Dynamic call forwarding-forwarding incoming calls without answering the call-is not supported on BroadWorks R20 server.

Enable Call Forward on Shared Lines

Parameter Function	template > parameter
Enable or disable server-based call forwarding per-registration. This parameter overrides `voIpProt.SIP.serverFeatureControl.cf`.	**reg-advanced.cfg** > reg.x.serverFeatureControl.cf
Enable or disable per-registration diversion on shared lines.	**sip-interop.cfg** > divert.x.sharedDisabled
Enable or disable server-based call forwarding. This parameter overrides `reg.x.serverFeatureControl.cf`.	**sip-interop.cfg** > voIpProt.SIP.serverFeatureControl.cf
This parameter depends on the value of `voIpProt.SIP.serverFeatureControl.cf`.	**sip-interop.cfg** > voIpProt.SIP.serverFeatureControl.localProcessing.cf
Enable or disable call forwarding behavior on all calls received. This parameter overrides `voIpProt.SIP.serverFeatureControl.localProcessing.cf`.	**sip-interop.cfg** > reg.x.serverFeatureControl.localProcessing.cf
Enable or disable the diversion feature for shared lines. This feature is disabled on most call servers.	**sip-interop.cfg** > call.shared.disableDivert

Private Hold on Shared Lines

Enable the private hold feature to display the PvtHold soft key on a shared line. When users in an active call on a shared line press the soft key, the active call is placed on hold and displays the shared line as busy to others sharing the line. The shared line also shows as busy when users transfer a call or initiate a

conference call during an active call. When you enable the feature, users can hold a call, transfer a call, or initiate a conference call and the shared line displays as busy to others sharing the line.

You can configure private hold only using configuration files; you cannot configure the feature on the Web Configuration Utility or from the local phone interface.

Note that `call.shared.exposeAutoHolds` is an existing parameter updated for private hold and `reg.X.enablePvtHoldSoftKey` is a new parameter for this feature.

Configure Private Hold

Parameter Function	template > parameter
Enable or disable the private hold feature for all lines.	**sip-interop.cfg** > call.shared.exposeAutoHolds
Enable or disable the Private Hold soft key for a specific shared line.	**reg-advanced.cfg** > reg.x.enablePvtHoldSoftKey

Enable Multiple Call Appearances

You can enable each registered phone line to support multiple concurrent calls and have each concurrent call display on the phone's screen. For example, you can place one call on hold, switch to another call on the same registered line, and have both calls display. You can set the maximum number of concurrent calls per registered line and the default number of calls per line key. If you assign a registered line to multiple line keys, the default number of concurrent calls applies to all line keys.

This feature is one of several features associated with flexible call appearances. If you want to enable multiple line keys per registration, see the section Assign Multiple Line Keys Per Registration.

Enable Multiple Call Appearances

Parameter Function	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

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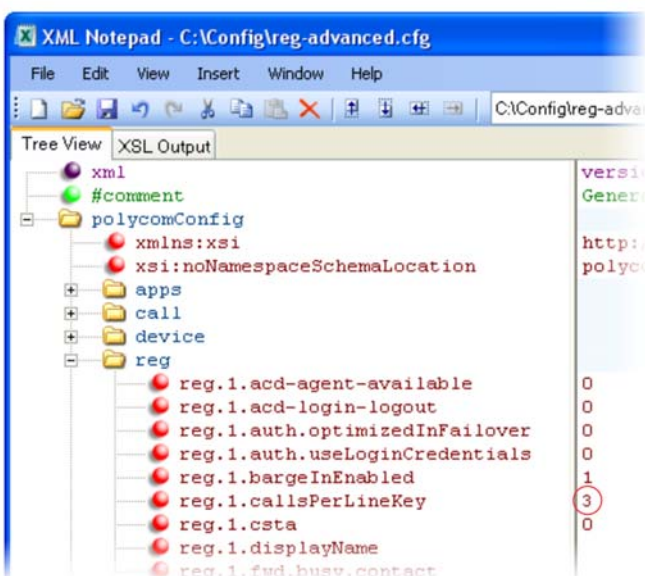
After you have set the `reg.1.callsPerLineKey` parameter to `3`, you can have three call appearances on line 1. By default, additional incoming calls are automatically forwarded to your voicemail. If you have more than two call appearances, a call appearance counter displays at the top-right corner of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

A number of features are associated with flexible call appearances. Use the following table to understand how you can organize registrations, line keys per registration, and concurrent calls per line key.

In the following table:

- **Registrations.** The maximum number of user registrations
- **Line Keys.** The maximum number of line keys
- **Line Keys Per Registration.** The maximum number of line keys per user registration
- **Calls Per Line Key.** The maximum number of concurrent calls per line key
- **Concurrent Calls (includes Conference Legs).** The runtime maximum number of concurrent calls. (The number of conference participants minus the moderator.)

Flexible Call Appearances

Phone Model	Registrations	Line Keys	Line keys Per Registration	Calls Per Line Key	Concurrent Calls*
VVX 101, 201	1	2	2	8	8 (2)

Flexible Call Appearances (continued)

VVX 300/301/310/311	34	48	34	24	24 (2)
VVX 400/401/410/411	34	48	34	24	24 (2)
VVX 500/501	34	48	12	24	24 (2)
VVX 600/601	34	48	12	24	24 (2)
VVX 1500	34	48	24	24	24 (2)
SoundStructure VOIP Interface **	12	12	12	24	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.

** For more information on using line and call appearances with the SoundStructure VOIP Interface, refer to the SoundStructure Design Guide, available at Polycom Support.

Enable Bridged Line Appearance

Bridged line appearance connects calls and lines to multiple phones. See the following table 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 the section Bridged Line Appearance Signaling.

> **Tip: Bridged line and shared call appearance 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

Parameter Function	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

Example Bridged Line Appearance Configuration

To begin using bridged line appearance, you must 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 the phone to subscribe with a SIP URL to a voicemail server contact. You can also configure the phone to access voicemail with a single key, for example, the **Messages** key on the VVX 300 series and 400 series phones, the **MSG** key on the VVX 1500 phone, and the **Messages** icon on the VVX 500 series and 600 series phones. 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 following table shows you the parameters you can configure.

Voicemail Integration

Parameter Function	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 phone screen backlight illuminates when you receive a new voicemail message.	**site.cfg** > mwi.backLight.disable

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.*
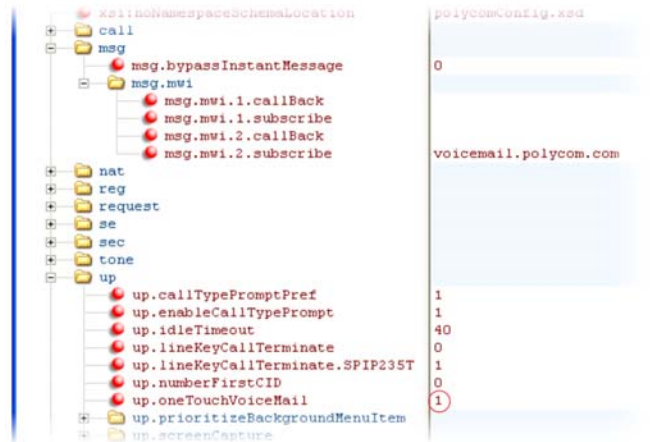
The following illustration shows that, in the **sip-basic.cfg** template, the default `callBackMode` setting for line 2 is set to `registration`. The phone uses the address assigned to line 2 to subscribe to the voicemail server you entered in `msg.mwi.2.subscribe`.
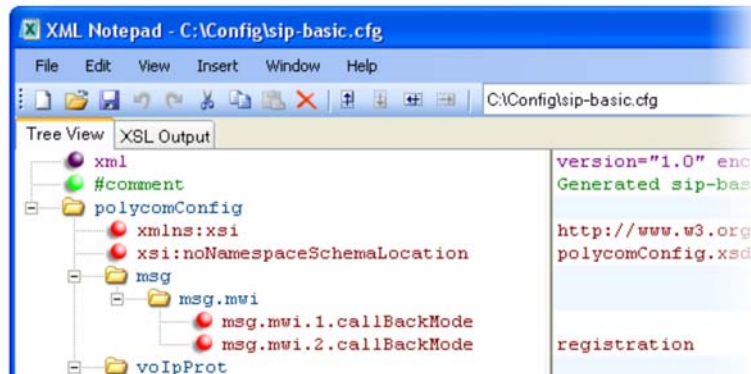
After this is enabled in the sip-interop.cfg template, on the phone, press the Messages key and select Message Center to access your voicemail.

Record and Play Audio Calls Locally

You can configure the VVX phones to record audio calls to a USB device that you plug into the phone. You can play back recorded audio on the phone as well as on other devices that run applications like Windows Media Player® or iTunes® on a Windows®- or Apple®-based computer.

To enable this feature, the USB device must be compatible with Polycom phones.

Web Info: Supported USB devices

For a list of supported USB devices, see S*upported USB Devices for Polycom SoundPoint IP 650 and VVX Phones: Technical Bulletin 38084* and *Supported USB Headsets for Polycom VVX 500 Business Media Phones: Engineering Advisories 62760* at Polycom Engineering Advisories and Technical Notifications.

You can enable call recording with the parameter shown in the table Record and Play Audio Calls. Audio calls are recorded in **.wav** format and include a date/time stamp, for example, **20Apr2007_190012.wav** was created on April 20, 2007, at 19:00:12. The phone displays the recording time remaining on the attached USB device and you can browse all recorded files using the phone's menu.

> **Note: Inform parties when you are recording calls**
> Federal, state, and/or local laws may legally require that you to notify some or all of the call parties that you are recording.

Record and Play Audio Calls

Parameter Function	template > parameter
To enable or disable call recording.	**features.cfg** > feature.callRecording.enabled

Example Call Recording Configuration

To record audio from the phone, you need a USB device plugged into the phone, and you need to enable the call recording feature in the **features.cfg** template file. In **features.cfg**, locate `feature.callRecording.enabled` and enter `1`, as shown next.
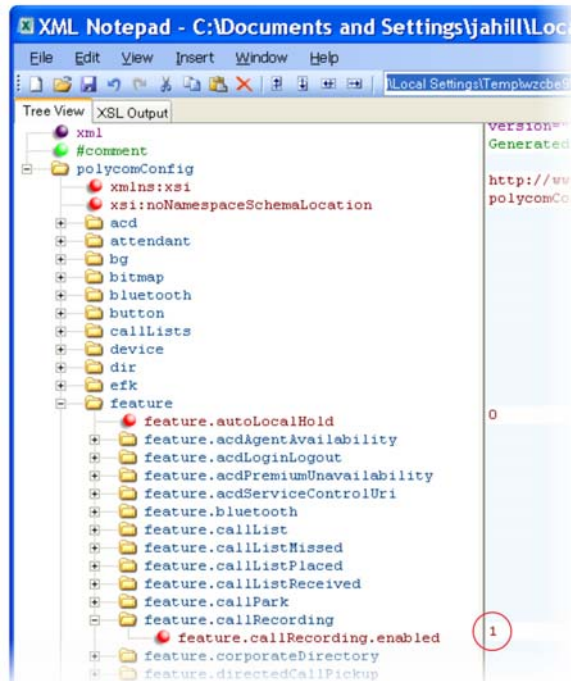
Press the **Pause** soft key to pause recording and press the **Stop** soft key to stop recording.

You can browse recorded audio files by navigating on the phone to **Menu** or **Settings > Removable Storage Media > Browse Recordings**.

Enable Centralized Call Recording

This feature, available on Polycom VVX phones, enables users to record audio and video calls and control call recording directly from phones registered with BroadSoft BroadWorks r20 server. Administrators must enable this feature on the BroadSoft BroadWorks r20 server and on the phones using the configuration parameters listed in the table Centralized Call Recording Parameters. On the BroadSoft server, administrators assign phone users one of several call recording modes listed in **Call Recording Modes**. You can manage your recorded audio and video files on a third-party call recording server.

> ⚠️ **Caution: Enable only one recording mechanism on the phone, not both**
> You can record calls using a central server or locally using the phone's USB call recording feature – you cannot use both at the same time. By default, both features are disabled. If you enable one call recording feature, ensure that the other is disabled. Use either centralized or the local call recording; do not use both.

By default, far-side participants are not alerted that calls are being recorded; BroadWorks server r20 provides administrators the option to enable an announcement at the beginning of a call that the call is being recorded. If a call being recorded is transferred, the new call continues to be recorded.

This feature can be enabled using the configuration parameters in the following table.

Centralized Call Recording Parameters

Parameter Function	template > parameter
Enable or disable the BroadSoft BroadWorks v20 call recording feature for all lines on a phone.	**sip-interop.cfg** > voIpProt.SIP.serverFeatureControl.callRecording
Enable or disable BroadSoft BroadWorks v20 call recording feature for a specific line on a phone.	**reg-advanced.cfg** > reg.x.serverFeatureControl.callRecording

Set the phone call recording modes on BroadSoft BroadWorks R20 server. You can set the following call recording modes:

- **Never Mode**   Call recording is never initiated and the phone never displays call recording soft keys.

- **Always Mode**   In Always mode, the entire incoming or outgoing call is recorded and no control options are available to users. During active calls, the phone displays a Record symbol. Call recording stops when the call ends and the call is stored on the server.

- **Always with Pause/Resume Support Mode**   Call recording starts automatically when the call connects and a Pause soft key displays. When you press the Pause soft key, a Resume soft key displays. The phone display indicates the status of the call recording state. Call recording stops when the call ends and the recorded part of the call is stored on the server.

- **On Demand Mode**   In On Demand mode, recording mode starts on the server when the call connects but the recorded file is not saved until you press the **Record > Start** soft key. When you press the Start soft key, the recording is saved to the server and the phone displays the Pause and Resume soft keys

- **On Demand Mode with User-Initiated Start Mode**   In On Demand with User-Initiated Start Mode, recording of a call does not begin automatically and a Record soft key displays. If you want to record during an active call, press Record to enter the recording sub-menu. Press the **Start** soft key to start recording and save to the server. While recording, the phone displays the Pause, Resume, and Stop soft keys.

● **Recording two separate calls and creating a conference**   This mode enables you to record two participants as separate calls sessions when connected in a conference call. The server stores the conference call as two separate recording sessions.

> **Troubleshooting: Record soft key does not display**
> When you select the Start or Pause soft key while recording an active call and the server sends an error processing your selection, the Record soft key does not display for the duration of the call. To display the Record soft key again, end and then reconnect the call.

Example Call Recording Configuration

This section provides an example configuration for the call recording feature.
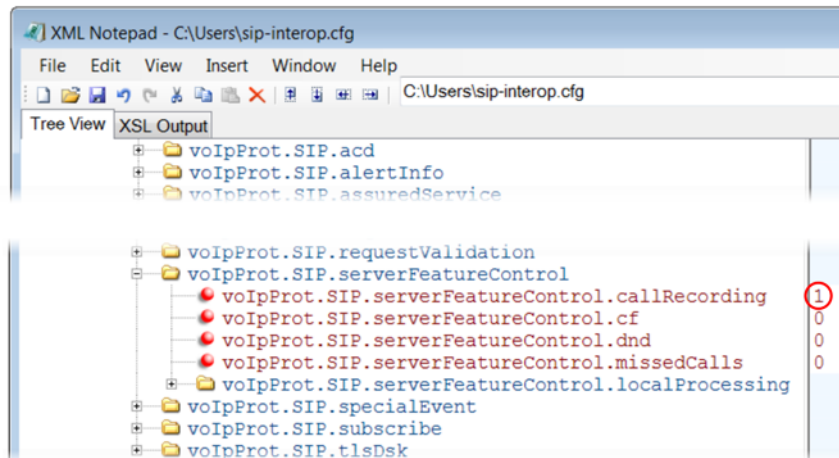
Use Busy Lamp Field

The busy lamp field (BLF) feature enables users to monitor the status of lines on remote phones, display remote party information, and answer incoming calls to remote phones (called directed call pickup). The BLF feature must be supported by a call server and the specific functions vary with the call server you use. You may need to consult your SIP server partner or Polycom channel partner to find out how to configure BLF.

Prior to UC Software 3.2 (Polycom SIP versions 2.1-3.1), a basic version of BLF was available on VVX 1500 phones. This basic BLF enables you to be monitored and to monitor idle and active phone states. As of UC Software 3.2, Polycom added enhanced BLF to all Polycom phones except VVX 1500 phones; this enhanced version also notifies you of the ringing state of a monitored phone. Currently, the enhanced version is available on all Polycom phones except VVX 1500 phones. You can monitor VVX 1500 phones and use VVX 1500 phones to monitor other phones; however, VVX 1500 phones monitoring other phones notify you of the idle state and active states of monitored phones and do not notify you of the ringing state. Note that BLF is not available with Polycom phones registered with Skype for Business Server.

The table Busy Lamp Field lists the parameters you may need to set. You can set up multiple BLF lines and monitor remote phones in active, ringing, and idle state. When BLF is enabled and you are monitoring a remote user, a BLF line key icon displays on the phone's screen. You can configure the line key label, and how call appearances and caller ID information are displayed. As of SIP 3.2.0, you can configure one-touch call park and retrieve and one-touch directed call pickup. Specifying the type of monitored resource as normal or automata changes the default actions of key presses. As the resource type, enter normal if the monitored resource type is a phone and automata if the monitored resource type is, for example, a call orbit.

If you select normal, pressing the BLF line key places an active call on hold before dialing the selected BLF phone. If you select automata, pressing the BLF line key immediately transfers active calls to that resource. To learn how to configure a park orbit and for examples, refer to the section Configure Enhanced Feature Keys.

Note that how you manage calls on BLF lines depends on the state of your phone—whether it is in the idle, active, or alerting state.

> **Web Info: Managing monitored lines**
> For information on how to manage calls to monitored phones, see the section Handling Remote Calls on Attendant Phones in *Using Statically Configured Busy Lamp Field with Polycom SoundPoint IP and VVX Phones: Technical Bulletin 62475 at Polycom Engineering Advisories and Technical Notifications.*

As of the SIP 3.1.0 release, the BLF feature was updated in the following ways:

- The phone gives a visual and audible indication when monitored BLF lines have incoming calls.
- The phone displays the caller ID of incoming calls to a remote monitored phones.

  BLF lines display a Pickup soft key that you can press to answer incoming calls to that monitored resource.

As of the SIP 3.2 release, the BLF feature was updated in the following ways:

- You can create a list of monitored parties to a maximum of 47 and configure the line key labels.
- You can configure key functions.
- You can disable spontaneous call appearances from incoming calls on monitored lines.

The following call servers are known to support this feature:

- Back to Back2 User Agent (B2BUA) Architecture
  - ➢ Metaswitch Metasphere Call Feature Server (CFS)
  - ➢ Asterisk® v1.6 or later
  - ➢ BroadSoft® BroadWorks
- Proxy Architecture
  - ➢ Avaya® SipX Enterprise Communications Server (ECS)
  - ➢ eZuce openUC™

These proxy architectures may support the full range of statically configured BLF features. However, they do not provide configuration control through their web management console.

The following call servers may support this feature, depending on the call server software variation and deployment:

- Proxy Architecture
  - ➢ OpenSIPS (formerly OpenSER)
  - ➢ Repro ReSIProcate

These proxy architectures or any other proxy server that allows the phone end-to-end communications with the monitored phone should be supported. However, these solutions have not been specifically tested by Polycom nor does Polycom guarantee their full interoperability.

> **Note: Use BLF with TCPpreferred transport**
> Use this feature with TCPpreferred transport (see <server/>).

Busy Lamp Field

Parameter Function	template > parameter
Specify an index number for the BLF resource.	**features.cfg** > attendant.reg
Specify the ringtone to play when a BLF dialog is in the offering state.	**features.cfg** > attendant.ringType
Specify the SIP URI of the call server resource list.	**features.cfg** > attendant.uri
Specify how call appearances and remote party caller ID display on the attendant phone.	**features.cfg** > attendant.behaviours.display.*
Specify the address of the monitored resource, a label for the resource, and the type of resource.	**features.cfg** > attendant.resourceList.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Example BLF Configuration

Typically, call servers support one of two methods of BLF configuration. Using the first method, you subscribe to a BLF resource list that is set up on your call server. Using the second method, you enter BLF resources to a configuration file and the call server directs the requests to those BLF resources. If you are unsure which method to use, consult your SIP server partner or Polycom Channel partner. This section shows you how to set up BLF using both methods.

To subscribe to a BLF list on a call server, you must access the call server and set up a list of monitored resources. The call server provides you with an address for that BLF resource list. To subscribe to that list, enter the address and any other information specific to your call server in the `attendant.uri` field located in the **features.cfg** template file, as shown next.
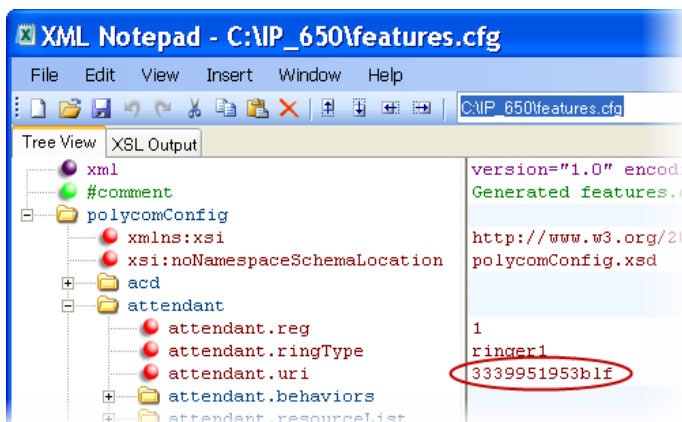
To specify BLF resources in the configuration file, open the **features.cfg** template file and enter the address (phone number) of the BLF resource you want to monitor, the label that displays beside the line key on the

phone, and the type of resource you are monitoring. Multiple registrations are available for a single SIP server. Your call server must support static BLF in order to configure BLF using the static method. In the following example, the phone is monitoring *Craig Blunt* and *Lucy Patterson*.
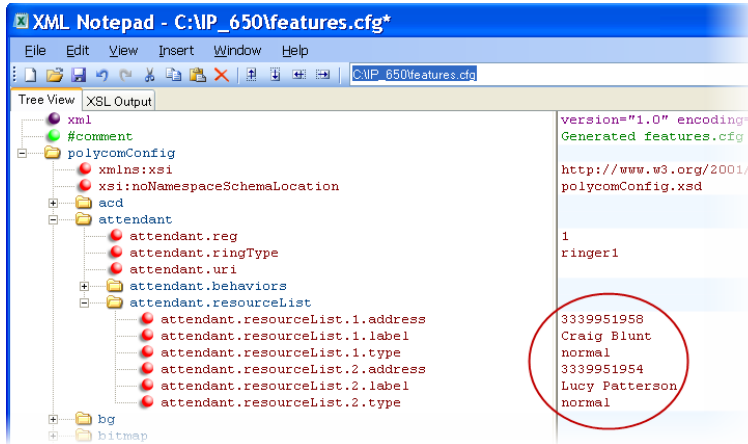
Both configuration methods result in the following BLF contacts—called BLF resources—beside line keys on the phone:
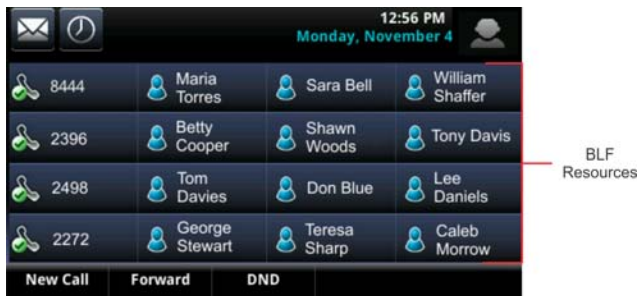
The following table shows the BLF key icons.

BLF Line Key Icons

States	Line Icons
Line monitoring is active	
Monitored line is busy	
Monitored line is ringing	

Configure Group Call Controls

This section provides information on configuring group call controls.

Enable Instant Messaging

All phones can send and receive instant text messages. Phones registered with Microsoft Skype for Business Server cannot send or receive instant messages. See the following table for the parameter you need to set to enable instant messaging. Once the feature is enabled, the phone's message waiting indicator (MWI) LED alerts you to incoming text messages visually; you can also set audio alerts. When you want to send an instant message, you can use the phone's dial pad to type your messages or you can choose a short message from a preset list. You can send instant messages by initiating a new dialogue or by replying to a received message. In addition, you can choose the message destination manually or you can select a contact from your local contact directory; see the section Use the Local Contact Directory.

Enable Instant Messaging

Parameter Function	template > parameter
Enable or disable instant messaging.	**features.cfg** > feature.messaging.enabled

Example Instant Messaging Configuration

The following illustration shows you how to enable instant messaging in the features.cfg template.
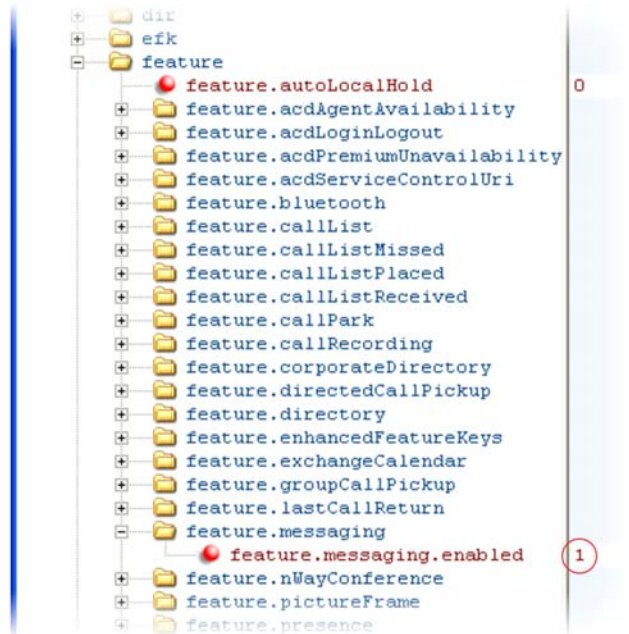
After setting this parameter, press the Messages key on the phone's keypad to display the Instant Messages option. Select the Instant Messages menu to send and receive instant messages.

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

Parameter Function	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 displays a **Pickup** soft key when you go off-hook. If you select **Pickup**, the **Group** soft key displays.

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.

Create Conferences

You can set up local or centralized audio and video conferences. Local conferences require a host phone to process the audio and video of all parties. Alternatively, you can use an external audio bridge, available via a central server, to create a centralized conference call. All Polycom phones support local- and server-based centralized conferencing. Polycom recommends using centralized conferencing for

conferences with four or more parties. The availability of centralized conferencing and features can vary by the call platform you use.

The maximum number of callers you can host in a local conference varies by phone:

- **VVX phones.** Support three-way calls
- **SoundStructure VoIP Interface.** Supports three-way calls

See the parameters in the following table 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 performs 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

Parameter Function	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. VVX phone users can choose which conference call participants to exchange video with.

Manage Conferences

Parameter Function	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.

```
        dir
    +   efk
    -   feature
            feature.autoLocalHold                    0
        +   feature.acdAgentAvailability
        +   feature.acdLoginLogout
        +   feature.acdPremiumUnavailability
        +   feature.acdServiceControlUri
        +   feature.bluetooth
        +   feature.callList
        +   feature.callListMissed
        +   feature.callListPlaced
        +   feature.callListReceived
        +   feature.callPark
        +   feature.callRecording
        +   feature.corporateDirectory
        +   feature.directedCallPickup
        +   feature.directory
        +   feature.enhancedFeatureKeys
        +   feature.exchangeCalendar
        +   feature.groupCallPickup
        +   feature.lastCallReturn
        +   feature.messaging
        -   feature.nWayConference
                feature.nWayConference.enabled       ①
        +   feature.pictureFrame
        +   feature.presence
```

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

Configure Intercom Calls

The Intercom feature enables users to place an intercom call that is answered automatically on the dialed contact's phone. This is a server-independent feature provided the server does not alter the Alert-Info header sent in the INVITE. You can configure the behavior of the answering phone using `voIpProt.SIP.alertInfo.x.class`.

The following procedure tells you how to place an intercom call when the intercom feature is enabled using configuration parameters.

To place an intercom call:

1   Press the **Intercom** soft key.

    The New Call screen displays.

2   In the **New Call** screen, enter a number or select a contact from the directory or call lists.

You have the option to initiate intercom calls using enhanced feature keys (EFKs). For information on configuring EFK functions, see the section Configure Enhanced Feature Keys. You do not need to disable the default Intercom soft key to create a custom soft key. For example, you can create an intercom action string:

- $FIntercom$

  This is an F type macro that behaves as a custom Intercom soft key. Pressing the soft key opens the Intercom dial prompt you can use to place an Intercom call by entering the destination's digits and using a speed dial or BLF button.

- <number>$Tintercom$

  This is a T type macro enables you to specify a Direct intercom button that always calls the number you specify in <number>. No other input is necessary.

In the following illustration, the action given as *0161$Tintercom*. When the Intercom soft key is pressed, an intercom call is placed to 0161. Because `softkey.3.insert` is set to 2, the Intercom soft key displays at the second position. However, for some features, soft key positions are fixed and in this example, PTT is enabled, which means a PTT soft key occupies a fixed second position and the Intercom soft key displays at the third position, as shown next.

Intercom Parameters

Parameter Function	template > parameter
Enable or disable the intercom feature.	**features.cfg** > feature.intercom.enable
Enable or disable the Intercom icon on the device home screen.	**features.cfg** > homescreen.intercom.enable
Enable or disable the intercom soft key.	**features.cfg** > softkey.feature.intercom
The string you want to use in the Alert-Info header.	**sip-interop.cfg** > voIpProt.SIP.intercom.alertInfo
A string to match the Alert-Info header in the incoming INVITE.	**sip-interop.cfg** > voIpProt.SIP.alertInfo.x.value
Specify a ring class name.	**sip-interop.cfg** > voIpProt.SIP.alertInfo.x.class

Configure Push-to-Talk and Group Paging

The push-to-talk (PTT) and group paging features are supported on all Polycom phone models installed with UC Software 4.0.0 or later.

The group paging feature enables you to make pages —one-way audio announcements—to users subscribed to a page group. The PTT feature is a collaborative tool that enables you to exchange broadcasts to users subscribed to a PTT channel, much like a walkie-talkie. You can transmit pages and PTT broadcasts using your handset, headset, or speakerphone and you can reject them, place them on

hold, and end them at any time. PTT broadcasts can be received on the speakerphone, handset, and headset, and pages can be received only through the speakerphone. Both features are available on all phones that use UC Software 4.0.0 or later.

You can enable one of these features or you can operate both simultaneously. There are 25 groups/channels you can subscribe to.

- **PTT Mode**—PTT mode is intended primarily for Wi-Fi phones. In PTT mode, the phone behaves like a walkie-talkie; you can broadcast audio to a PTT channel and recipients subscribed to that channel can respond to your message. To configure PTT, see the table Configure Push-to-Talk.

- **Paging Mode**—Paging mode is intended primarily for desktop phones. in paging mode, you can send announcements to recipients subscribed to a page group. In page mode, announcements play only through the phone's speakerphone. To configure paging, see the table Configure Group Paging.

Administrators must enable paging and PTT before users can subscribe to a page group or PTT channel.

Web Info: Using a different IP multicast address

The push-to-talk and group paging features use 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.

Push-to-Talk

You specify the same IP multicast address in the parameter ptt.address for both PTT and paging mode. PTT administrator settings are located in the site.cfg template file. The parameters shown in the following table are located in the features.cfg template file.

Configure Push-to-Talk

Parameter Function	template > parameter
Specify the IP multicast address used for the PTT and paging features.	**site.cfg** > ptt.address
Enable PTT mode.	**site.cfg** > ptt.pageMode.enable
Specify the name to display (per phone).	**site.cfg** > ptt.pageMode.displayName
Change default settings for PTT mode.	**site.cfg** > ptt.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Group Paging

You specify the same IP multicast address in the parameter ptt.address for both PTT and paging mode. Paging administrator settings shown in the following table are located in the site.cfg template file. Page group settings are located in the features.cfg template file.

Configure Group Paging

Parameter Function	template > parameter
Specify the IP multicast address used for the PTT and paging features.	**site.cfg** > ptt.address

Configure Group Paging

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.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Web Info: Configuring push-to-talk and group paging

Though the example configurations in this section get you started, Polycom recommends that you read how to use PTT and paging in the *Polycom VVX Business Media Phones User Guide* before configuring settings.

Example PTT/Paging Configuration

The following illustration shows the default PTT and paging administrator settings in the site.cfg template file.

Note that you can enter a display name for sent PTT broadcasts in ptt.displayName and for sent page announcements in ptt.pageMode.displayName. The two following illustrations show the range of PTT channels and page groups you can subscribe to.

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

PTT Mode Channels

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.

Paging Mode Groups

Use Hoteling

The hoteling feature enables users to use any available shared phone by logging in to a guest profile. The following table shows you the parameters you can configure. After logging in, users have access to their own guest profile and settings on the shared phone. To use Hoteling, you must configure Polycom phones with the BroadSoft BroadWorks R17 platform and use UC Software 4.0.2 or later. This feature is available on Polycom VVX 300 series, 400 series, 500 series, 600 series, and 1500 phones.

> **Web Info: Use the hoteling feature**
> For details on configuring the hoteling feature, see *Using Hoteling on Polycom Phones*: *Feature Profile 76554* at Polycom Engineering Advisories and Technical Notifications.

You can use hoteling in conjunction with the feature-synchronized automatic call distribution (ACD) feature. For information, see the section Configure Feature-Synchronized Automatic Call Distribution (ACD).

Use Hoteling

Parameter Function	template > parameter
Enable or disable hoteling.	**features.cfg** > feature.hoteling.enabled
Choose a line registration index.	**features.cfg** > hoteling.reg

Example Hoteling Configuration

This example configuration shows the hoteling feature enabled and uses registration line 1. In the features.cfg template, the feature.hoteling.enabled parameter is set to 1 to enable.
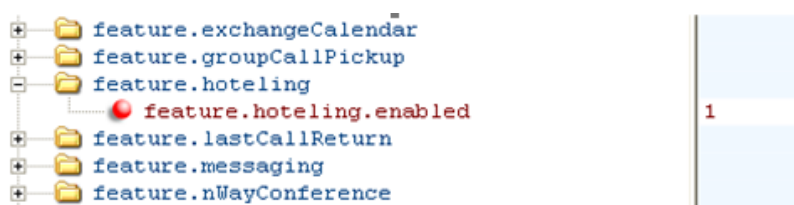
The hoteling feature is applied to phone line 1.
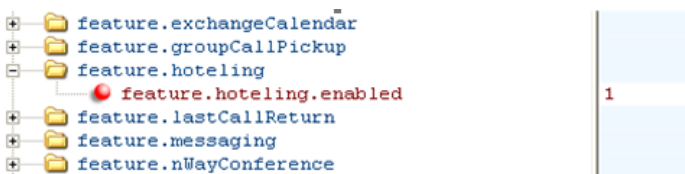
When hoteling is enabled, the line 1 index key *2326* has hoteling enabled and the **GuestIn** soft key displays.

Configure SIP-B Automatic Call Distribution

You can use your VVX business media phones in a call center agent/supervisor role on a supported call server.

Configure SIP-B Automatic Call Distribution

Parameter Function	template > parameter
To turn Automatic Call Distribution on or off.	**features.cfg** > feature.acdLoginLogout.enabled
To enable or disable Automatic Call Distribution for a specific registration.	**reg-advanced.cfg** > reg.x.acd-login-logout
To enable or disable Feature Synchronized ACD.	**sip-interop.cfg** > voIpProt.SIP.acd.signalingMethod

The also support ACD agent availability. This feature depends on support from a SIP server.

ACD Agent Availability

Parameter Function	template > parameter
To turn ACD Agent Availability on or off.	**features.cfg** > feature.acdAgentAvailable.enabled
To enable or disable ACD Agent Availability feature for a specific registration.	**reg-advanced.cfg** > reg.x.acd-agent-available

Example SIP-B Automatic Call Distribution Configuration

In the following illustration, in the **reg-basic.cfg** template file, three line registrations and labels have been set up.
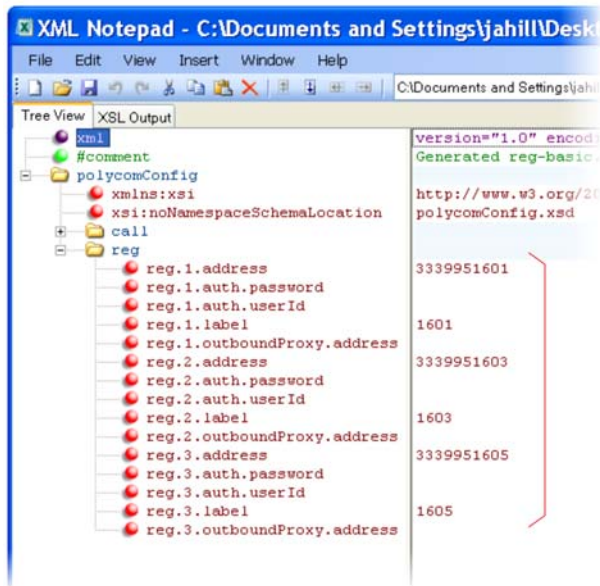
In this example, SIP-B ACD is enabled in **features.cfg** using the parameters feature.acdAgentAvailability.enabled and feature.acdLoginLogout.enabled, as shown next.
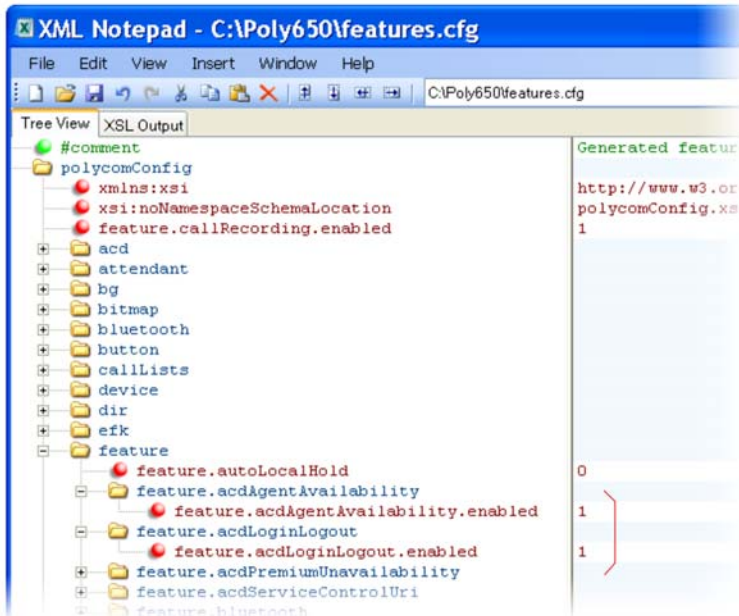
You must also enable SIP-B ACD in the **reg-advanced.cfg** template file. The next illustration shows the two parameters you need to enable to display the ACD soft keys on the phone screen.
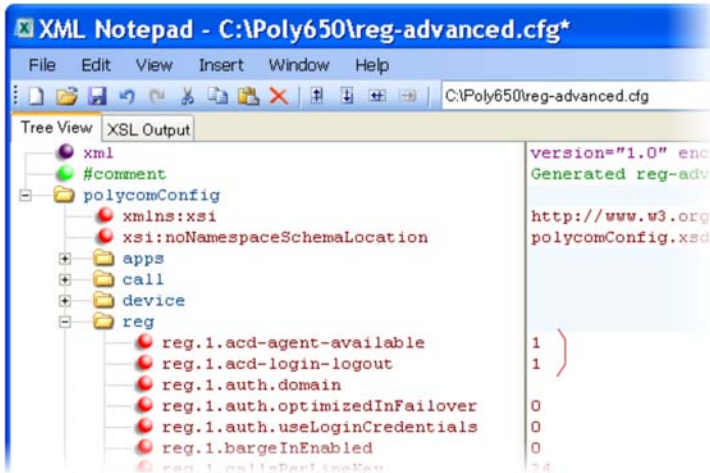
Once SIP-B ACD is enabled, the following soft keys display on the phone.

The ACD agent 1601 displays on phone line 1 and the agent can log in and out of the ACD feature.

Configure Feature-Synchronized Automatic Call Distribution (ACD)

You can use your VVX phones in a call center agent/supervisor role on a supported call server. Feature-synchronized ACD is distinct from and provides more advanced ACD functions than the Hoteling feature (see Use Hoteling).

Feature-synchronized automatic call distribution (ACD) enables organizations that handle a large number of incoming phone calls to use in a call center role. Feature-synchronized ACD is available as a standard or a premium service. The premium ACD service has been enhanced in two ways: *Hoteling* and *Queue Status Notification*. Hoteling enables agents to use their agent credentials to log in to any available phone. If you want to use the hoteling feature with feature-synchronized ACD, see the section Use Hoteling. Queue status notification enables agents to view the queue status of a call center so that agents can adjust their call response.

> **Web Info: Further information on ACD enhancements**
>
> For more information on standard and premium ACD as well as the hoteling and queue status notification enhancements, see *Using Premium Automatic Call Distribution for Call Centers: Feature Profile 76179* at Polycom Engineering Advisories and Technical Notifications.

See the following table for parameters you can configure. When standard functions are enabled, the phone indicates it is in the ACD call center agent state. Phone users can sign in and sign out of the ACD state as a call center agent using soft keys or the phone's menu. When ACD is enabled and a user is signed in as an agent, the phone can display the current state of the agent, for example, whether the agent is available or unavailable to take new calls.

The capabilities of this feature vary with the SIP call server. Please consult your call server provider for information and for documentation. The SIP signaling used for this implementation is described in the BroadSoft BroadWorks document *Device Key Synchronization Requirements Document; Release R14 sp2; Document version 1.6*.

The following phones support the feature-synchronized ACD feature:

● VVX 300 series, 400 series, 500 series, 600 series, and 1500 business media phones. Note that you must use UC Software 5.0 or later to use this feature with the VVX business media phones.

Configure Feature Synchronized Automatic Call Distribution

Parameter Function	template > parameter
To turn Feature Synchronized ACD on or off.	**features.cfg** > feature.acdLoginLogout.enabled
To turn ACD Agent Availability on or off.	**features.cfg** > feature.acdAgentAvailable.enabled
To turn Premium Feature Synchronized ACD on or off.	**features.cfg** > feature.acdPremiumUnavailability.enabled
To turn Feature Synchronized ACD Control URI on or off.	**features.cfg** > feature.acdServiceControlUri.enabled
To set the registration to be used for Feature Synchronized ACD and the users' sign-in state.	**features.cfg** > acd.*
To enable or disable Feature Synchronized ACD.	**sip-interop.cfg** > voIpProt.SIP.acd.signalingMethod

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Example Feature Synchronized ACD Configuration

In the following illustration, in the **reg-basic.cfg** template file, three line registrations and labels have been set up.
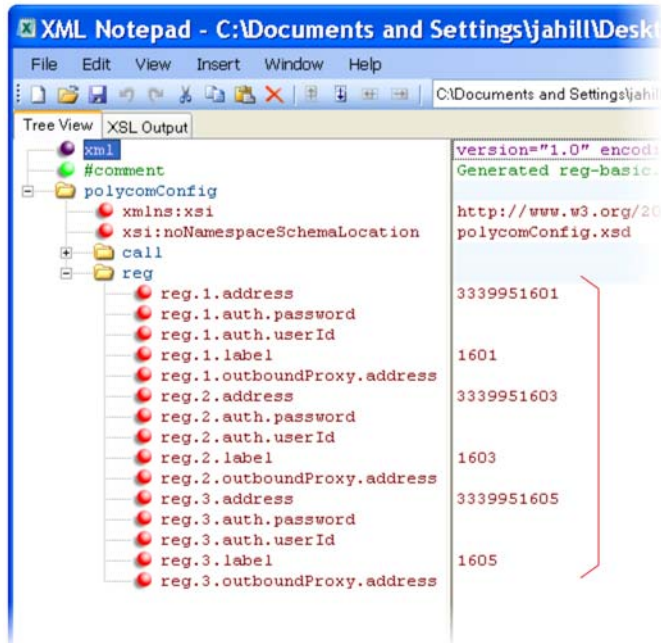
To enable feature-synchronized ACD for these registrations, in the sip-interop.cfg template file, set `voIpProt.SIP.acd.signalingMethod=1,` as shown next.
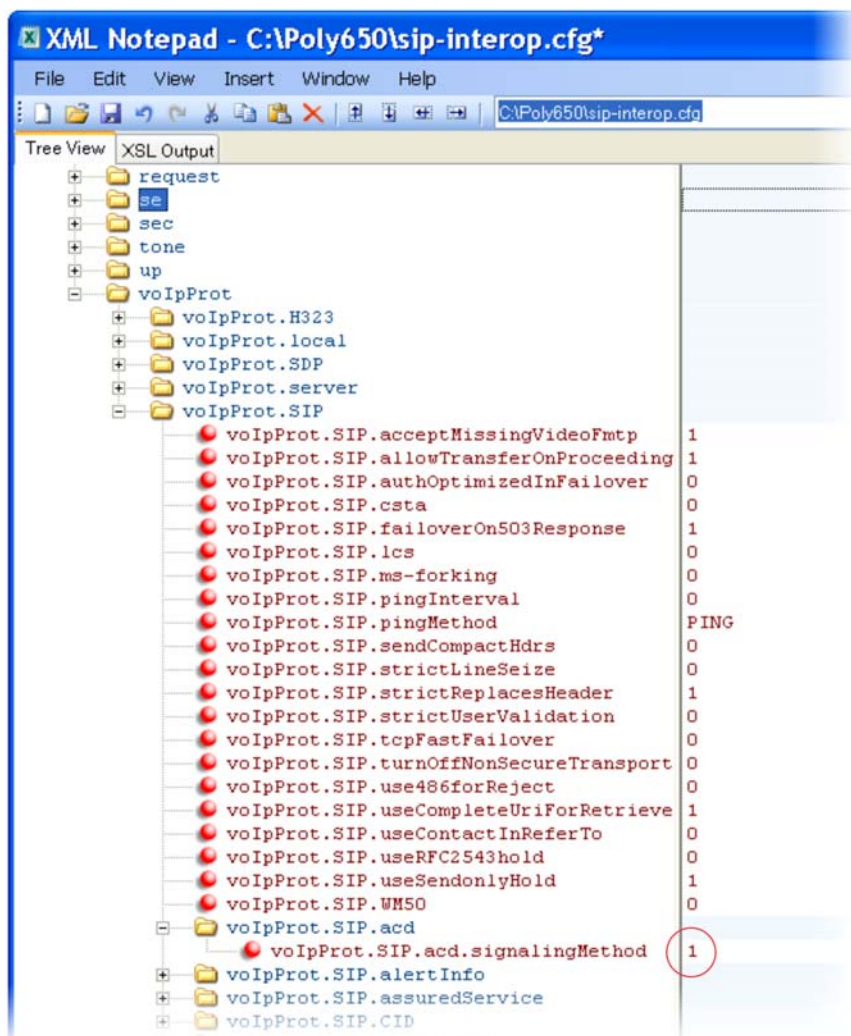
A shown next, you must enable the feature.enhancedFeaturekeys.enabled parameter, four parameters in feature.acd*, and the acd.reg and acd.stateAtSignIn parameters. If you want to use reason codes, enable acd.unavailreason.active and enter the reason codes in the acd.x.unavailreason.codeName and acd.x.unavailreason.codeValue parameters. You can define up to 100 reason codes. In the following example, two reason codes have been enabled and set to Out to lunch and On the phone.
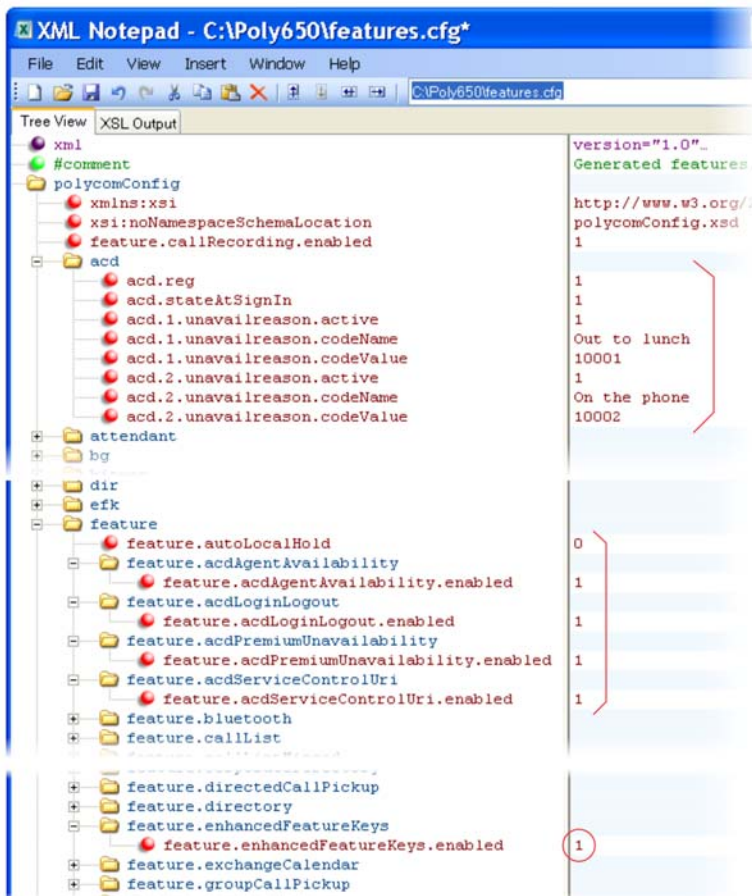
The ACD agent number displays on the screen and the available status soft keys display: New Call, Forward, Unavailable. When the agent presses the Unavailable soft key, the reason codes you entered display for the agent to select.

Web Info: Configuration details for feature-synchronized ACD

For details on how to configure phones for feature-synchronized ACD, see *Using Feature-Synchronized Automatic Call Distribution with Polycom Phones: Feature Profile 57216* at Polycom Engineering Advisories and Technical Notifications.

Customize Phone Functions

This section shows you how to set up a number of custom phone functions.

Configure TR-069

Technical Report-069 (TR-069) is a technical specification that provides a mechanism for remote management of end-user devices. TR-069 as a bidirectional SOAP/HTTP-based protocol enables secure communication between Auto Configuration Servers (ACS) and Polycom phones. Using TR-069, Polycom phones can be remotely configured and managed by provisioning systems that comply with TR-069 technical specification.

You can configure the TR-069 feature through the phone menu, Web Configuration Utility, or configuration parameters on a central server. You can configure Polycom phones with an ACS server, including user name and password using DHCP Option 43 for IPv4 and DHCP Option 17 for IPv6.

Polycom provides parameters for the TR-104 and TR-106 data models that support provisioning of TR-069-enabled devices by an Auto-Configuration Server (ACS). TR-104 is a parameter data model for VoIP-only devices, and TR-106 is a parameter data model for all TR-069-enabled devices.

The following Polycom VVX Business Media Phones support TR-069:

- VVX 101
- VVX 201
- VVX 3xx
- VVX 4xx
- VVX 5xx
- VVX 6xx
- VVX 1500

Configure TR-069 Settings on the Phone Menu

You can configure TR-069 settings on the phone menu.

To configure TR-069 settings on the phone menu:

1 Go to **Settings > Advanced > Administration Settings > Network Configuration**.
2 Enable **TR-069**.
3 In the**TR069 Menu**, go to **ACS Configuration** and configure the following:
    ➢ URL
    ➢ Username
    ➢ Password
    ➢ Periodic Inform
    ➢ Inform Interval
4 In **Phone/CPE Configuration**, configure a user name and password.
5 In **Upgrade Management**, select **Enable** or **Disable**.

Configure TR-069 from the Web Configuration Utility

You can configure TR-069 from the Web Configuration Utility.

To configure the TR-069 from the Web Configuration Utility:

» Go to **Settings > Provisioning Server > TR-069 Menu**.

Configure TR-069 using Centralized Provisioning

You can configure TR-069 using centralized provisioning. The following table lists parameters you can configure.

TR-069 Parameters

Parameter Template	Permitted Values
`device.feature.tr069.enabled` tr069.cfg	0 (default) - Disables TR-069 feature 1 - Enables TR-069 feature
`device.tr069.acs.password` tr069.cfg	Sets the TR-069 ACS server password used to authenticate the phone. Null (default) String (256 maximum characters)
`device.tr069.acs.url` tr069.cfg	Sets the URL for the TR-069 ACS server. Null (default) URL (256 maximum characters)
`device.tr069.acs.username` tr069.cfg	Sets the TR-069 ACS server username used to authenticate the phone. PlcmSpip (default) String (256 maximum characters)
`device.tr069.cpe.password` tr069.cfg	Specifies the TR-069 CPE password, which authenticates a connection request from the ACS server. Null (default) String (256 maximum characters)
`device.tr069.cpe.username` tr069.cfg	Specifies the TR-069 CPE user name, which authenticates a connection request from the ACS server. PlcmSpip (default) String (256 maximum characters)
`device.tr069.periodicInform.enabled` tr069.cfg	Indicates whether the CPE must periodically send CPE information to ACS using the Inform method call. 0 (default) - Periodic Inform call is disabled. 1 - Periodic Inform call is enabled.

TR-069 Parameters

Parameter Template	Permitted Values
`device.tr069.periodicInform.interval` tr069.cfg	Specifies the time interval in seconds in which the CPE must attempt to connect with the ACS to send CPE information if `device.tr069.periodicInform.enabled=1`. 18000 (default) 0 to 36000
`device.tr069.upgradesManaged.enabled` tr069.cfg	Indicates whether the ACS manages image upgrades for the phone or not. 0 (default) - The phone uses ACS or provisioning server for upgrade. 1 - The phone upgrades only from the ACS server.
`log.level.change.tr069` tr069.cfg	Sets the log levels for the TR-069 feature. 4 (default) 0 - 6

Map TR-106 Parameters to Polycom Parameters

The data model TR-106 defines the TR-069 ACS parameter details. Parameters listed as 'Internal Value' are not directly mapped to a configuration parameter on the phone and the phone generates these values dynamically to provide to the ACS server.

The following table lists the TR-106 parameters and their corresponding Polycom parameters.

TR-106 Parameters to Polycom Parameters

TR-106 ACS parameter names	Parameter (Polycom parameter names)	Writable
Device		
Device.DeviceInfo		
Manufacturer	Internal Value	No
ManufacturerOUI	Internal Value	No
ModelName	Internal Value	No
ProductClass	Internal Value	No
SerialNumber	Internal Value	No
HardwareVersion	Internal Value	No
SoftwareVersion	Internal Value	No
UpTime	Internal Value	No
Device.ManagementServer.		

TR-106 Parameters to Polycom Parameters

TR-106 ACS parameter names	Parameter (Polycom parameter names)	Writable
URL	device.tr069.acs.url	Yes
Username	device.tr069.acs.username	Yes
Password	device.tr069.acs.password	Yes
PeriodicInformEnable	device.tr069.periodicInform.enabled	Yes
PeriodicInformInterval	device.tr069.periodicInform.interval	Yes
ConnectionRequestURL	Internal Value	No
ConnectionRequestUsername	device.tr069.cpe.username	Yes
ConnectionRequestPassword	device.tr069.cpe.password	Yes
UpgradesManaged	device.tr069.upgradesManaged.enabled	Yes
STUNServerAddress	tcpIpApp.ice.stun.server	Yes
STUNServerPort	tcpIpApp.ice.stun.udpPort	Yes
STUNUsername	tcpIpApp.ice.username	Yes
STUNPassword	tcpIpApp.ice.password	Yes
Device.LAN.		
IPAddress	Internal Value	No
SubnetMask	Internal Value	No
DNSServers	Internal Value	No
MACAddress	Internal Value	No
MACAddressOverride	Internal Value	No

Map TR-104 Parameters to Polycom Parameters

The data model TR-104 defines the TR-069 ACS parameter details. Parameters listed as 'Internal Value' are not directly mapped to a configuration parameter on the phone and the phone generates these values dynamically to provide to the ACS server.

The following table list the TR-104 parameters and their corresponding Polycom parameters.

TR-104 Parameters to Polycom Parameters

TR-104 ACS parameter names	CPE Parameter (Polycom parameter names)	Writable
VoiceService.{i}.VoiceProfile.{i}.		
DigitMap	dialplan.digitmap	Yes
VoiceService.{i}.VoiceProfile.{i}.SIP.		

TR-104 Parameters to Polycom Parameters

TR-104 ACS parameter names	CPE Parameter (Polycom parameter names)	Writable
RegistrarServer	voIpProt.server.X.address	Yes
RegistrarServerPort	voIpProt.server.X.port	Yes
OutboundProxy	voIpProt.SIP.outboundProxy.address	Yes
OutboundProxyPort	voIpProt.SIP.outboundProxy.port	Yes
RegisterExpires	voIpProt.server.X.expires	Yes
RegistersMinExpires	voIpProt.server.X.expires.overlap	Yes
RegisterRetryInterval	voIpProt.server.X.retryTimeOut	Yes
VoiceService.{i}.VoiceProfile.{i}.SIP.EventSubscribe.{i}.		
ExpireTime	voIpProt.server.X.subscribe.expires	Yes
VoiceService.{i}.VoiceProfile.{i}.H323.		
Gatekeeper	voIpProt.server.H323.X.address	Yes
GatekeeperPort	voIpProt.server.H323.X.port	Yes
VoiceService.{i}.VoiceProfile.{i}.RTP.		
LocalPortMin	tcpIpApp.port.rtp.mediaPortRangeStart	Yes
LocalPortMax	tcpIpApp.port.rtp.mediaPortRangeEnd	Yes
VoiceService.{i}.VoiceProfile.{i}.RTP.SRTP.		
Enable	sec.srtp.enable	Yes
VoiceService.{i}.VoiceProfile.{i}.ButtonMap.Button.{i}.		
ButtonName	softkey.X.label	Yes
FacilityAction	softkey.X.action	Yes
UserAccess	softkey.X.enable	Yes
VoiceService.{i}.VoiceProfile.{i}.Line.{i}.		
DirectoryNumber	reg.X.address	Yes
VoiceService.{i}.VoiceProfile.{i}.Line.{i}.SIP.		
AuthUserName	reg.X.auth.userId	Yes
AuthPassword	reg.X.auth.password	Yes
VoiceService.{i}.VoiceProfile.{i}.Line.{i}.CallingFeatures.		
CallForwardUnconditionalEnable	reg.X.fwdStatus	Yes
CallForwardUnconditionalNumber	reg.X.fwdContact	Yes

TR-104 Parameters to Polycom Parameters

TR-104 ACS parameter names	CPE Parameter (Polycom parameter names)	Writable
CallForwardOnBusyEnable	reg.X.fwd.busy.status	Yes
CallForwardOnBusyNumber	reg.X.fwd.busy.contact	Yes
CallForwardOnNoAnswerEnable	reg.X.fwd.noanswer.status	Yes
CallForwardOnNoAnswerNumber	reg.X.fwd.noanswer.contact	Yes
CallForwardOnNoAnswerRingCount	reg.X.fwd.noanswer.ringCount	Yes
DoNotDisturbEnable	divert.dnd.X.enabled	Yes

Supported TR-069 Remote Procedure Call (RPC) Methods

The following table lists the supported RPC methods.

RPC Methods

RPC Method	Description
GetRPCMethods	Discovers the set of methods supported by the phone.
SetParameterValues	Modifies the value of one or more phone parameters.
GetParameterValues	Obtains the value of one or more phone parameters.
GetParameterNames	Discovers the parameters accessible on a particular phone.
GetParameterAttributes	Reads the attributes associated with one or more phone parameters.
SetParameterAttributes	Modifies attributes associated with one or more phone parameters.
Reboot	Reboots the phone.
Download	Causes the phone to download a specified file from the designated location. Supported file types for download: Firmware Image Configuration File
FactoryReset	Resets the phone to its factory default state.
TransferComplete	Informs the ACS of the completion (either successful or unsuccessful) of a file transfer initiated by an earlier Download or Upload method call.
AddObject	Adds a new instance of an object defined on the phone.
DeleteObject	Removes a particular instance of an object.

Lock the Basic Settings Menu

By default, all users can access the Basic settings menu available on the RealPresence Trio 8800 system and VVX business media phones. From this menu, users can customize non-administrative features on

their phone. You can choose to lock the Basic settings menu and only allow certain users access to the menu to customize the phone.

If enabled, you can use the default user password (123) or administrator password (456) to access the Basic settings menu, unless the default passwords are not in use.

Lock the Basic Settings Menu

Parameter Function	template > parameter
Require a password to access the Basic settings menu on the phone.	**features.cfg** > up.basicSettingsPasswordEnabled

Configure the Microbrowser and Web Browser

The VVX business media phones support a full web browser and microbrowser. The VVX 101 business media phone does not support a microbrowser. The functions and performance of the microbrowser and web browser vary by phone model.

You can configure the microbrowser and web browser to display a non-interactive web page on the phone's idle screen, and you can specify an interactive home web page that you can launch in a web browser. When you select a link that displays on the idle browser the phone launches that link in the web browser.

If the browser uses over 30MB of memory and either the amount of free memory on the phone is below 6MB or the real time is between 1AM to 5AM, the browser restarts. After the browser has restarted, the last displayed web page is restored.

On the VVX 1500 phone, you can launch the web browser by pressing the **App** key on the phone or by navigating to **Menu > Applications**.

On the VVX 201, 3xx series, 4xx series, 5xx series, and 6xx series phones, go to **Home > Applications**.

Polycom provides a default microbrowser and browser feature for the phone's idle screen. My Info Portal is a Polycom-developed application that gives you access to the latest news, sports, weather, stock, and other news. You can sign up for access to My Info Portal through the Polycom VVX phone or through a computer. The first time you sign in to My Info Portal, you are asked to accept the Polycom End User Licensing Agreement (EULA).

On VVX 50x and 60x series business media phones, you can use the parameter `up.OffHookIdleBrowserView.enabled` to enable the microbrowser and associated soft keys to continue to display on the phone when the phone goes off-hook. The microbrowser will continue to display until the user enters a number. Typically, the microbrowser only displays when the phone is idle and not in a call.

To display My Info Portal on the VVX idle browser, set:
- `mb.idleDisplay.home="http://idle.myinfoportal.apps.polycom.com/idle"`
  OR
- `mb.main.home="http://myinfoportal.apps.polycom.com"`
  AND
- `mb.idleDisplay.refresh="600"`

For more information, see the P*olycom Web Application Developer's Guide* at Polycom UC Software Support Center.

Microbrowser and Web Browser Parameters

Parameter Function	template > parameter
Specify the Application browser home page, a proxy to use, and size limits.	**applications.cfg** > mb.*
Specify the Telephony Event Notification events to be recorded and the URL where notifications are sent.	**applications.cfg** > apps.telNotification.*
Specify phone state polling settings, such as response mode, the poll URL, and a user name and password.	**applications.cfg** > apps.statePolling.*
Specify the push server settings, including message type, port, tunnel, and a user name and password.	**applications.cfg** > apps.push.*
Specify whether or not the idle browser displays on the phone screen when the phone goes off-hook.	**sip-interop.cfg** > up.OffHookIdleBrowserView.enabled

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Example Microbrowser and Web Browser Configuration

The following example shows you how to set a web page on the idle screen of the VVX phone and how to set the interactive web browser's home page on the VVX 1500 phone.
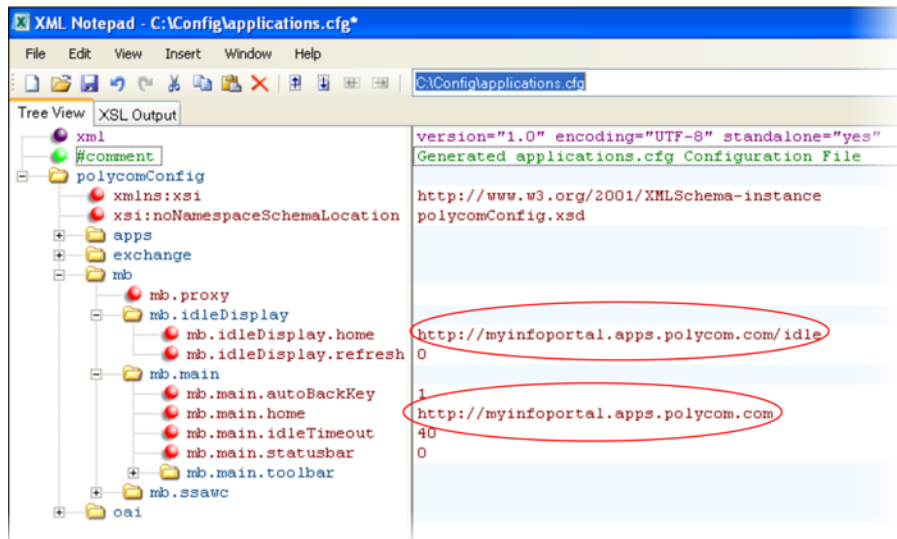
The following illustration shows a non-interactive idle web browser on the VVX 1500 phone.

The following illustration shows the web browser's interactive home page on the VVX 1500 phone.

Configure Soft Keys

You can customize the functions of the phone's soft keys. This feature is typically used to access frequently used functions, to create menu shortcuts to frequently used phone settings; or, if your phone does not have a particular hard key, you can 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. 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 Understand Macro Definitions. You can configure soft keys on all VVX phones.

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 for 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.

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

Note that if you are using UC Software 5.1.0 on VVX phones, you can disable the display of any default soft key to make room for custom soft keys; you cannot disable default soft keys with any other UC Software release. 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 following table 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 Configure Enhanced Feature Keys for details about configuring soft keys and line keys on the phone.

Configure Soft Keys

Parameter Function	**template** > parameter
To turn enhanced feature keys on (required).	**features.cfg** > feature.enhancedFeatureKeys.enabled

Configure Soft Keys (continued)

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 displays in.	**features.cfg** > softkey.x.use.*
To display soft keys for various phone features, including default soft keys.	**features.cfg** > softkey.feature.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Example Soft Key Configurations

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

> **Web Info: Using configurable soft keys**
>
> For more example configurations, see the two following documents at Polycom Engineerig Advisories and Technical Notifications:
> - *Using Enhanced Feature Keys and Configurable Soft Keys on Polycom Phones: Technical Bulletin 42250*
> - *Using Enhanced Feature Keys (EFK) Macros to Change Soft Key Functions on Polycom Community: Feature Profile 42250*

Example 1

Use the following example to automatically transfer an active call to BroadSoft voicemail. In this example, *55 is the star code for BroadSoft voicemail, and 8545 is the extension of the voicemail line the call transfers to. Note that enabling the parameter `softkey.1.use.active` causes the soft key to display when a call becomes active on the line. When you press the soft key—labelled 'VMail' in this example—the call is placed on hold and automatically transferred to BroadSoft voicemail.

To map a send-to-voicemail enhanced feature key sequence to a soft key:

1 Update the configuration file as follows:

   ➢ `softkey.1.label="VMail"`
   ➢ `softkey.1.action="$FTransfer$$Cpause1$$FDialpadStar$$FDialpad5$$FDialpad5$$FDialpad8$$FDialpad5$$FDialpad4$$FDialpad5$$FSoftKey1$"`
   ➢ `softkey.1.enable="1"`
   ➢ `softkey.1.use.active="1"`

2 Reboot the phone.

   When an incoming call connects and becomes active, the **VMail** soft key displays.

3 Press the **VMail** soft key to transfer an incoming call to voicemail.

Example 2

The following example enables you to enter a voicemail extension to transfer an active call to BroadSoft voicemail. In this example, *55 is the star code for BroadSoft voicemail. Note that enabling the parameter `softkey.1.use.active` causes the soft key to display when a call becomes active on the line. When you press the soft key, the call is placed on hold and a field prompts you to enter the extension of the voicemail line you want to transfer the call to. The `efk.prompt*` parameters control the numeric prompt field you enter the extension to. Note that this example works only on line 1 of your phone.

To create a send-to-voicemail prompt that allows a mailbox number entry:

1 Update the configuration file as follows:

   ➢ `softkey.1.label="VMail"`
   ➢ `softkey.1.action="^*55$P1N10$$Tinvite$"`
   ➢ `softkey.1.enable="1"`
   ➢ `softkey.1.use.active="1"`
   ➢ `efk.efkprompt.1.label="Voice Mail"`
   ➢ `efk.efkprompt.1.status="1"`
   ➢ `efk.efkprompt.1.type="numeric"`

2 Reboot the phone.

   When an incoming call connects and becomes active, the **VMail** soft key displays.

3 Press the **VMail** soft key.

   A field displays prompting you to enter an extension.

4 Dial *55 and the extension you want to transfer the call to.

Example 3

This section provides an example of a speed dial soft key and an example of a speed dial line key linked to a directory file. In both example, the macro action includes a pause in the dialing sequence.

Use the following example to configure a soft key to automatically dial a number with a pause in the dialing sequence. In this example, use `$CpauseX$` where `X` is the number of seconds to pause—7 in this example. Adding this pause function enables you to automatically dial into a conference ID that requires an entry code after the conference call is connected.

To program a pause into a soft key dial number:

1 Update the configuration file as follows:

   ➢ `softkey.1.label="VMail"`
   ➢ `softkey.1.action="$S1$$Tinvite$$Cwc$$Cpause7$$FDialpad8$$FDialpad5$$FDialpad4$$FDialpad5$"`
   ➢ `softkey.1.enable="1"`
   ➢ `softkey.1.use.idle="1"`
   ➢ `feature.enhancedFeatureKeys.enabled="1"`

The values for this example are explained as follows:

- `$S1$`   Speed dial line 1
- `$S1$$Tinvite$$`   The phone sends an invite to $S1$
- `$Cwc$`   The phone waits for the call to connect
- `$Cpause7$`   The phone waits for 7 seconds before dialing the remaining numbers
- `$FDialpad8$$FDialpad5$$FDialpad4$$FDialpad5$`   The phone enters the entry code 8545

Use the following example to add a speed dial line key from a directory file. The speed dial includes a pause.

To program a pause into a directory-linked speed dial line key:

1 Update the configuration file as follows:

 ➢ `feature.enhancedFeatureKeys.enabled="1"`

 ➢ `efk.efklist.1.action.string="501$Tinvite$$Cwc$$Cpause7$1234#$Tdtmf$"`
    `efk.efklist.1.label="number"`

 ➢ `efk.efklist.1.mname="number"`

 ➢ `efk.efklist.1.status="1"`

2 In a contact directory file or speed dial file (000000000000-directory.xml or *<MACaddress>*-directory.xml), add the following:

 ➢ `<fn>Call Number</fn>`

 ➢ `<ct>!number</ct>`

 ➢ `<sd>99</sd>`

In the action string: `<ct>"501$Tinvite$$Cwc$$Cpause7$1234#$Tdtmf$"`:

- `501$Tinvite$`   Dial 501
- `$Cwc$`   Wait for the call to connect
- `$Cpause7$`   A seven second pause
- `1234#$Tdtmf$`   Send 1234 dual-tone multi-frequency

The EFK commands are linked to the directory file as follows:

- The parameter `efk.efklist.1.mname="number"` is linked to the speed dial contact `<ct>!number</ct>` of the directory file
- Use `<fn>Call Number</fn>` to define the name that displays on the key
- Use `<sd>99</sd>` to identify which directory entry to link to the key

> **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.

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 following table 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 <efk/>.

Web Info: Using enhanced feature keys

For instructions and details on how to use enhanced feature keys, see *Using Enhanced Feature Keys and Configurable Soft Keys on Polycom Phones: Technical Bulletin 42250.* at Polycom Engineerig Advisories and Technical Notifications.

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 EFK feature from SIP 3.0 is compatible with the EFK feature from SIP 3.1. However, improvements have been made and Polycom recommends that existing configuration files be reviewed and updated.

Enhanced Feature Keys

Parameter Function	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 you need to 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. The template directory configuration file is named **000000000000-directory~.xml**. To use this file, remove the tilde (~) from the file name. For more detailed information on using the contact directory, see Use the Local Contact Directory.

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Some Guidelines for Configuring Enhanced Feature Keys

The following guidelines 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 is used and the subsequent ones is 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 "^" to 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 after 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.
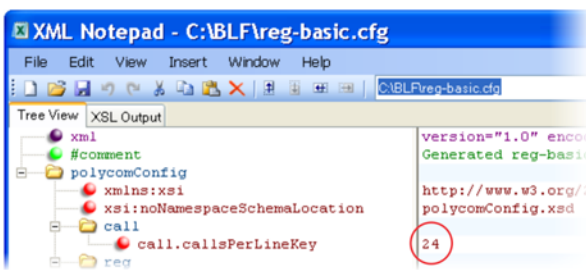
Enable the enhanced feature keys feature in the features.cfg template file, as shown next.

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 displays 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 are able to see the numbers you enter into the prompt.
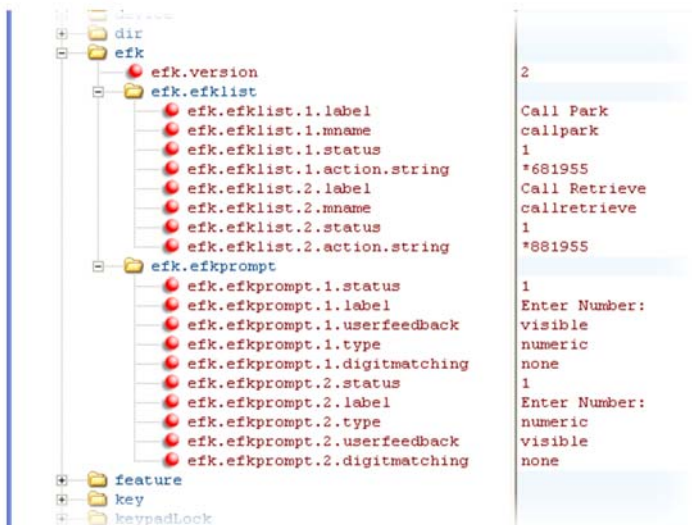
Understand Macro Definitions

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

- Macro Actions
- Prompt Macro Substitution
- Expanded Macros

- Special Characters

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 following table.

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 is 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`

`intercom`

Note: Polycom recommends that you always define this field. If it is not defined, the supplied digits are 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 Map Internal Key Functions.

Macro Actions and Descriptions

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 fails. 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 after 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 Macros

The action string
`$Changup$*444*$P1N4$$Tinvite$$Cwaitconnect$$P2N3$$Cpause2$$Tdtmf$$Changup$`is
executed in order as follows:

1. The user is prompted for 4 digits. For example, *1234*.
2. The user is prompted for 3 digits. For example, *567*.
3. The user's active call is disconnected.
4. The string **444*1234* is sent using the INVITE method.
5. After connection, there is a 2 second pause, and then the string *567* is sent using DTMF dialing on the active call.

6 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 (ct)` 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.
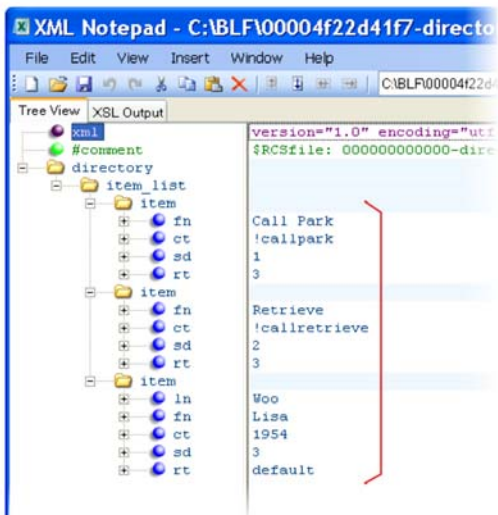
The following illustrates the call park and call retrieve line keys and a speed dial contact Lisa Woo.
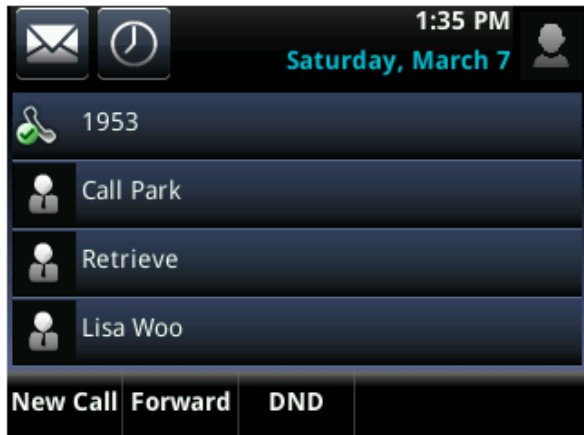
For an explanation of all fields in the directory file, see the table Understanding the Local Contact Directory.

Flexible Line Key Assignment

You can give your phone users the ability to assign a line key function to a line key anywhere on the phone's screen. Normally, functions are assigned line keys in succession, the order in which the line key displays on the phone. This feature enables you to break that ordering and assign a line key function to a line key that displays anywhere on the phone's screen. This feature is available on the VVX 300 series, 400 series, 500 series, 600 series, and VVX Expansion Modules. Refer to the table Flexible Line Key Assignment for the parameters you need to configure to set up this feature.

You can apply this feature to any line key function including line appearance, speed dial, busy lamp field (BLF), and presence. Line keys that you configure using this feature override the default line key assignments as well as any custom line key configurations you may have made. To use this feature, you need to specify the function of each line key on the phone. You do this by assigning a category and an index to each line key, both of which are explained in the example configuration.

Specific conditions apply when you assign busy lamp field (BLF) or presence to line keys. If you are assigning BLF or presence to a line key, assign that line key to index=0 to indicate automatic ordering. BLF and presence line keys are self-ordering, meaning that if you have these features assigned to multiple line keys, you can specify the location of the BLF or presence line key but not the order in which they display. For example, you can assign a BLF line key to index 1, 3, and 5 but you cannot specify how the contacts are ordered, which BLF contacts display on line keys 1, 3, and 5. In addition, to assign BLF and presence to a line key, you need to assign a corresponding registration line. You can configure multiple line keys per registration if each line key has a corresponding reg.x.lineKeys parameter.

Flexible Line Key Assignment

Parameter Function	template > parameter
To enable flexible line key assignment.	**reg-advanced.cfg** > lineKey.reassignment.enabled
Specify the line key category.	**reg-advanced.cfg** > lineKey.x.category
Specify the line key number (dependent on category).	**reg-advanced.cfg** > lineKey.x.index

Note: Line keys are numbered sequentially

Line keys on VVX phones and expansion modules are numbered sequentially, and the line keys on your expansion module depends on how many lines your phone supports. For example, a VVX 600/601 phone supports 16 lines, numbered 1-16. The first line on an expansion module connected to a VVX 600/601 phone is line 17.

Example Flexible Line Key Assignment Configuration

To enable flexible line key assignment, in the features.cfg template, set the lineKey.reassignment.enabled parameter to 1. Then assign each line key a category and an index. The category specifies the function of the line key and can include: Unassigned, Line, BLF, Speed Dial, and Presence. Note that the category Unassigned leaves that line key blank. The index specifies the order in which the line keys display on the phone screen. Use the following table to help you assign a category and an index to the line keys on your phone.

Assigning Flexible Line Keys

Assigning a Category and an Index to Line Keys					
Category	unassigned	line	BLF	Speed Dial	Presence
Index	Null	The line index number	0	The speed dial index number	0

The following illustration shows you an example flexible line key assignment configuration in the features.cfg template file:
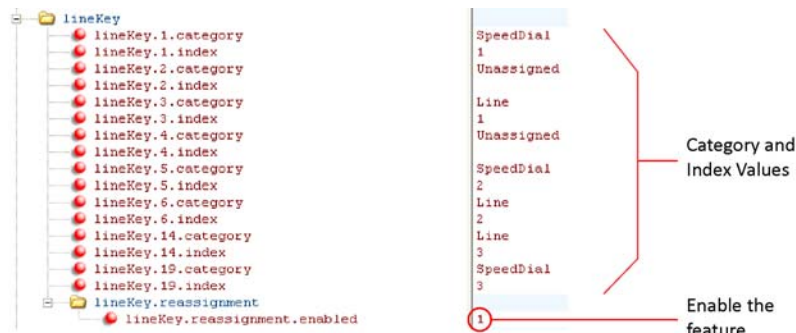
This configuration displays on a VVX 600/601 phone screen as the following:
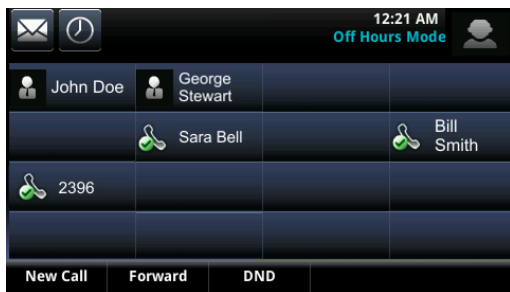
Configure the Phone Keypad

You can customize many of the default key functions on the phone's keypad interface. The table Configure Phone Keys lists the parameters you can configure to change the layout of your phone's keypad. Polycom recommends that you configure only those phone keys with removable key caps, including Directories, Applications, Conference, Transfer, Redial, Menu, Messages, Do Not Disturb, and Call Lists.

> **Caution: Choosing keys to remap**
>
> Polycom recommends that you remap only those keys with removable key caps. If you remap other keys, your phone may not work properly. You should not remap the following keys: the dial pad, volume control, handsfree, mute, headset, hold, and the navigation arrow keys.

You can configure phone keys in the following ways:

- You can assign function or features to a key.
- You can turn a phone key into a speed dial.
- You can assign enhanced feature key (EFK) operations to a phone key. For example, you can reach a phone menu path to a single key press using a macro code. To find out how to configure EFK functions, see the section Configure Enhanced Feature Keys.
- You can delete all functions and features from a phone key.

Configure Phone Keys

Parameter Function	template > parameter
Set the primary key function for key y on phone model x.	**features.cfg** > key.x.function.prim
Set the secondary key function for key y on phone model x.	**features.cfg** > key.x.subPoint.prim

Configure Phone Logs and Directory Files

This section provides information on configuring phone logs and phone directory files.

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.

As of Polycom UC Software 4.0.1, the VVX 500/501, and 1500 phones remember the previous call history after a restart or reboot.

> **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

Parameter Function	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 following table 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**
Indicates 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**

Call Log Elements and Attributes

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.

Use the Local Contact Directory

Polycom phones feature a contact directory file you can use to store frequently used contacts as a template contact directory file named **000000000000-directory~.xml** included with the UC Software download. This template file is loaded to the provisioning server the first time you boot up a phone with UC Software or when you reset the phone to factory default settings.

When you first boot the phone out of the box or when you reset the phone to factory default settings, the phone looks for contact directories in the following order:

- An internally stored local directory
- A personal **<MACaddress>-directory.xml** file
- A global **000000000000-directory.xml** file when the phone substitutes <000000000000> for its own MAC address.

To create a per-phone, personal directory file, replace *<000000000000>* in the global file name with the phone's MAC address: *<MACaddress>*-**directory.xml**. Any changes users make to the contact directory from the phone are stored on the phone drive and uploaded to the provisioning server in the personal directory (*<MACaddress>*-**directory.xml**) file, which enables you to preserve a contact directory during reboots.

To create a global directory file that you can use to maintain the directory for all phones from the provisioning server, remove the tilde (~) from the template file name **000000000000-directory.xml**. When you update the global directory file on the provisioning server, the updates are downloaded onto the phone and combined with the phone specific directory.

Using the parameter `voIpProt.SIP.specialEvent.checkSync.downloadDirectory`, you can configure the phones to download the updated directory files upon receipt of a `checksync NOTIFY` message. The files are downloaded when the phone restarts, reboots, or when the phone downloads any software or configuration updates.

Any changes to either the global or personal directory files are reflected in the directory on the phone after a restart or a `checksync NOTIFY` message. When merging the two files, the personal directory always takes precedence over the changes in the global directory. Thus, if a user modifies a contact from the global directory, the contact is saved in the personal directory file, and the contact from the global directory is ignored when the files are next uploaded.

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**, delete the *<MACaddress>*directory.xml and reset the phone to factory defaults.

> **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 file read only.

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 Apply Distinctive Incoming Call Treatment), presence (see Configure Presence), and instant messaging (see Enable Instant Messaging). The following table lists the directory parameters you can configure.

The following table lists the maximum number of contacts and maximum file size of the local Contact Directory for each phone. To conserve phone memory, use the parameter `dir.local.contacts.maxNum` to configure the phones to support a lower maximum number of contacts.

Maximum File Size and Number of Contacts

Phone	Maximum File Size	Maximum Number of Contacts in File
VVX 300/301/310/311	4MB	500
VVX 400/401/410/411	4MB	500
VVX 500/501 and 600/601	4MB	500
VVX 1500	102400 bytes Non-volatile: 100KB	9999
SoundStructure VoIP Interface	Not applicable	Not applicable

Use the Local Contact Directory

Parameter Function	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

Use the Local Contact Directory

Specify whether you are prompted for an Admin/User password when adding, editing, or deleting contacts in the Contact Directory.	**features.cfg** > dir.local.passwordProtected
Specify when the phone downloads the updated global directory file.	**feature.cfg >** voIpProt.SIP.specialEvent.checkSync.download Directory
The template contact directory file.	000000000000-directory~.xml

Example Configuration

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

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

Note: GENBAND contact duplication
You can duplicate contacts in the Contact Directory registered with the GENBAND server.

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]**

Understanding the Local Contact Directory

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.

lb	Label	UTF-8 encoded string of up to 40 bytes[1]

The label for the contact. 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 form the label. A space is added between first and last names.

Note: For GENBAND, the **Label** element is shown as **Nick Name**, and is a mandatory, non-duplicate field.

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 to 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.

Understanding the Local Contact Directory

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, automatically calls 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 is placed. The parameter for each of these options is outlined in the following table. 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 *Changes to Local Digit Maps on SoundPoint IP, SoundStation IP, and Polycom VVX 1500 Phones: Technical Bulletin 11572* at Polycom Engineering Advisories and Technical Notifications.

Configure the Local Digit Map

Parameter Function	**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 to used for routing calls.	**site.cfg** > dialplan.routing.server.x.*
Configure the same parameters as above for a specific registration (overrides the global parameters).	**site.cfg** > dialplan.x.*

Configure the Local Digit Map

Specifies the time in seconds that the phone waits before dialing a number when you dial on-hook.	**site.cfg** > dialplanuserDialtimeOut

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Polycom support for digit map rules varies for open SIP servers and Microsoft Skype for Business Server.

Use Open SIP Digit Map

The following is a list of digit map string rules for open SIP environments. 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 to 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 *Rs* with the digits between the last two *Rs*. For example, **R555R604R** 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|33xxH`—All four digit numbers starting with a *0* are placed using the SIP protocol, whereas all four digit numbers starting with *33* are placed using the H.323 protocol.

> **Note: Only VVX 500/501, 600/601, and 1500 phones match the HH**
>
> Only VVX 500/510, 600/611, and 1500 phones match the H. On all other phones, the H is ignored and users need to press the Send soft key to complete dialing. For example, if the digit map is 33xxH, the result is as follows:
>
> If a VVX 1500 user dials 3302 on an H.323 or dual protocol line, the call is placed after the user dials the last digit.

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 is 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 Rs or 5 Rs) is considered an invalid digit map.

- If you use T in the left part of *RRR's* syntax, the digit map will not work. For example, *R0TR322R* will not work.

Generate Secondary Dial Tone with Digit Maps

You can regenerate dial tone by adding a comma "**,**" to the digit map. In the following example, you can dial seven-digit numbers after dialing "8" as shown next in the rule **8,[2-9]xxxxxxT:**

[2-9]11|0T|011xxx.T|[0-1][2-9]xxxxxxxxx|8,**[2-9]xxxxxxT**|[2-9]xx.T

By adding digit "8", dial tone plays again and users can complete the remaining seven-digit number. In this example, if you also have a 4-digit extension that begins with "8" then you will hear dial tone after the first "8" was dialed because "8" matches the "8" in the digit map.

If you want to generate dial tone without the need to send the "8", replace one string with another using the special character "R" as shown next in the rule **R8RR**. In the following example, replace "8" with an empty string to dial the seven-digit number:

[2-9]11|0T|011xxx.T|[0-1][2-9]xxxxxxxxx|R8RR,[2-9]xxxxxxT|[2-9]xx.T

Configure the Speed Dial Feature

You can link entries in your local contact directory to speed dial contacts on the phone Home screen that enable you to place calls quickly using dedicated speed dial buttons. To set up speed dial through the phone's contact directory, see the section Use the Local Contact Directory. To set up speed dial contacts, you need to become familiar with parameters in the following table, which identifies the directory XML file and the parameters you need to set up your speed dial contacts.

You can assign contacts the following speed dial index ranges.

Speed Dial Index Ranges

Phone Model	Range
VVX 101, 201	1 – 99
VVX 300 series, 400 series, 500 series, and 600 series	1 – 500
VVX 1500	1 – 9999 The maximum number may be limited by the phone's available memory.
SoundStructure VoIP Interface	Not applicable.

On some call servers, enabling presence for an active speed dial contact displays that contact's status on the speed dial's line key label. For information on how to enable presence for contacts, see the section Configure Presence.

Configure the Speed Dial Feature

Parameter Function	**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**

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 boot up a phone with UC Software or when you reset the phone to factory default settings, 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 restart, reboot, or reset to factory defaults, 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 **<00000000000>** in the global file name with the phone's MAC address, for example, **<MACaddress>-directory.xml**.

When you reset the phone to factory defaults, the phone looks first for its own **<MACaddress>-directory.xml** and then 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.

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

After 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.
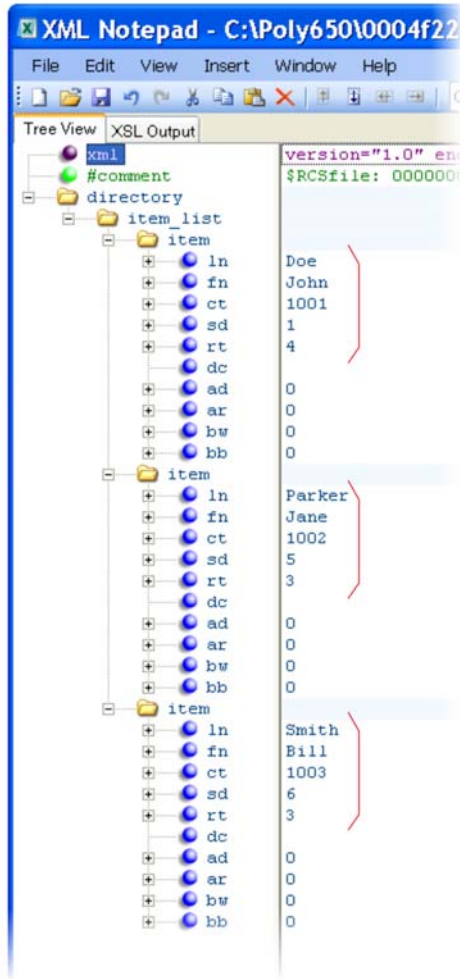
This configuration results in the following speed dial keys on the phone.
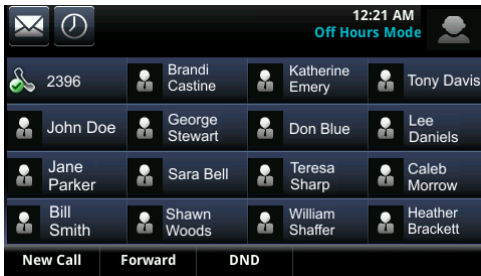
Use the Corporate Directory

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. After 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.

Polycom phones currently support 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

Polycom phones support 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 *Best Practices When Using Corporate Directory on Polycom Phones*: *Technical Bulletin 41137* at Polycom Engineering Advisories and Technical Notifications.

Use the Corporate Directory

Parameter Function	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.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Example Corporate Directory Configuration

The following example is a representation of the minimum parameters you must set to begin using the corporate directory. The exact parameters and values you 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.
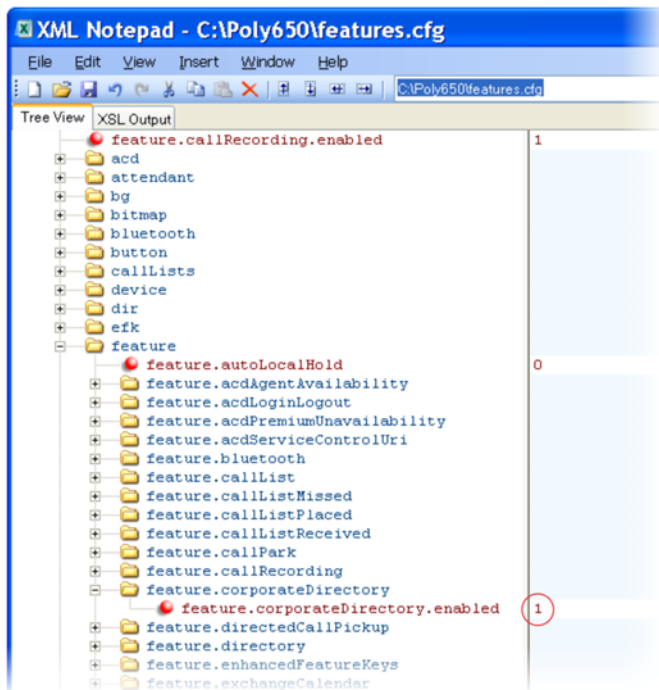
The following illustration points you to the minimum parameters you need to set. You need to enter a corporate directory address in dir.corp.address and specify where on the corporate directory server you

want to make queries in dir.corp.baseDN. In addition, you require a user name and password. The dir.corp.attribute.x.name must match the attributes in the server.
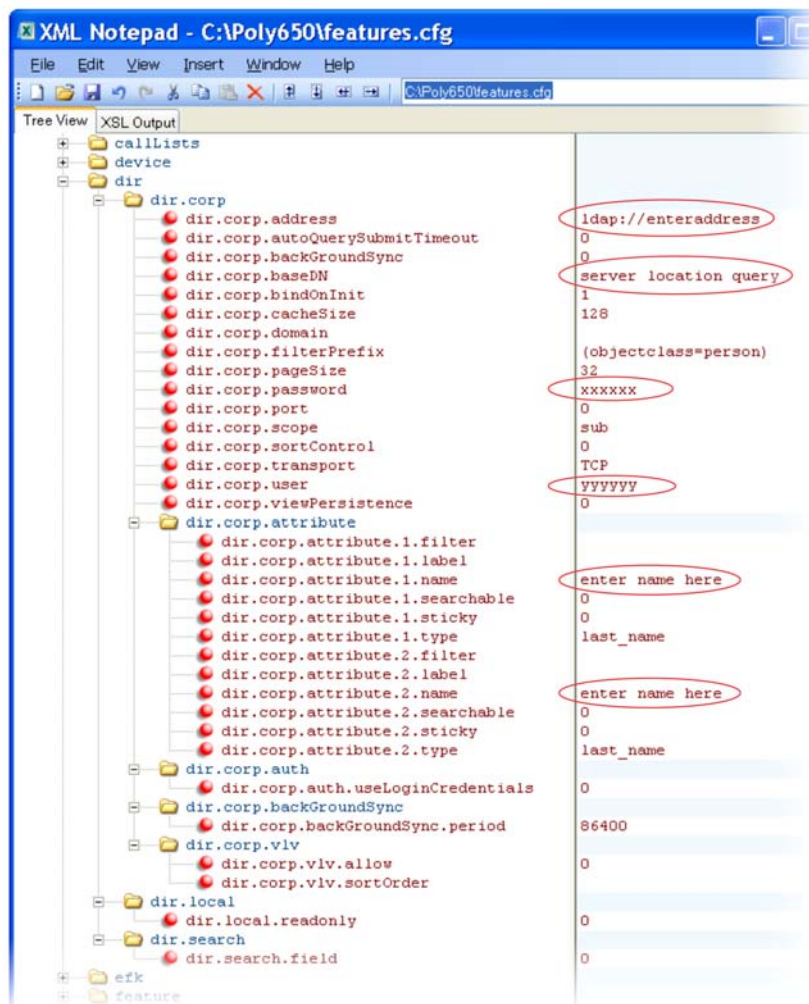
To search the corporate directory, press Directories on the phone and select Corporate Directory, as shown next.

Use Third-Party Servers

This section provides information on configuring phones and features with third-party servers.

Configure Polycom Phones with Alcatel-Lucent

This section shows you how to configure Polycom phones with Alcatel-Lucent (ALU) CTS.

Initiate and Manage Alcatel-Lucent Advanced Conferences

When you are signed into the ALU CTS, you can initiate ad-hoc conference calls with two or more contacts from your phone. You can also create a participant list and manage conference participants. Note you you can have only one active conference call in progress at a time on your phone. This feature is not supported on VVX 101 and 201 phones.

You can configure the number of participants in a conference using the parameter `reg.x.advancedConference.maxParticipants`. The number of participants you configure must match the number of participants allowed on the ALU CTS.

Advance Conference includes the following features:

* **Roster**   Provides a list of participants in the conference
* **Conference Controller**   The person who creates the conference and can add or drop participants, and mute and unmute participants.
* **Push-to-Conference**   Includes participants list when creating a conference call.
* Join two calls into a conference call
* Join a call to an active call to make a conference call

Configure ALU Advanced Conferences

Parameter Function	template > parameter
Enable or disable advanced conferences and conference controls.	**features.cfg** > feature.advancedConference.enabled
Enable or disable push-to-conference functionality.	**features.cfg** > reg.x.advancedConference.pushToConference
Specify the maximum number of participants allowed in a push to conference.	**features.cfg** > reg.x.advancedConference.maxParticipants
Enable or disable conference participants to receive notifications for conference events.	**features.cfg** > reg.x.advancedConference.subscribeForConfEvents
Enable or disable the conference host to receive notifications for conference events.	**features.cfg** > reg.x.advancedConference.subscribeForConfEventsOnCCPE

Barge In and Bridge In for ALU CTS

The Barge In feature is for BLF shared lines, and Bridge In is for Shared Call Appearance lines. This feature enables multiple users in a Shared Call Appearance group to view and bridge into active calls on a shared line. This feature is not supported on VVX 1500 business media phones.

By default, group members can bridge into active calls only. Users cannot bridge into held or incoming calls. Multiple people can bridge into one active call. This feature is disabled by default. You can enable this feature using the parameter reg.x.bridgeInEnabled.

Configure Barge In and Bridge In

Parameter Function	template > parameter
Enables or disables the Bridge In feature.	**features.cfg** > reg.x.bridgeInEnabled

Barge-In for Busy Lamp Field Lines

This feature enables users to barge in on active and held calls on Busy Lamp Field (BLF) lines and supports three barge-in modes: Normal, Whisper and Silent. This feature is not supported on VVX 1500 business media phones. The Barge In feature for BLF lines is disabled by default. You can enable the barge-in feature, the default barge-in mode, and whether or not a tone plays when a contact barges in on a call.

Configure Barge In for BLF

Parameter Function	template > parameter
Enable or disable barge-in and choose the default barge-in mode.	**features.cfg** > attendant.resourceList.x.bargeInMode
Enable or disable a tone that plays when a contact barges in on a call.	**features.cfg** > attendant.resourceList.x.requestSilentBargeIn

Dual Tone Multi Frequency (DTMF) Relay

This feature enables users to press DTMF commands during active SIP audio calls and conference calls to perform actions. This feature is not supported for H.323 calls.

Configure DTMF Relay

Parameter Function	template > parameter
Enable or disable DTMF relays for active SIP calls.	voIpProt.SIP.dtmfViaSignaling.rfc2976
Controls the behavior of the Star and Pound keys used for DTMF relays for active SIP calls.	voIpProt.SIP.dtmfViaSignaling.rfc2976.nonLegacyEncoding

Shared Call Appearance

The Shared Call Appearance feature enables users who share a line to monitor and bridge into calls on the shared line. Each line supports up to 21 call appearances. This feature is disabled by default. You can enable the feature and configure the hold request for the line. For ALU CTS, this feature is supported on VVX 300 series, VVX 400 series, VVX 500 series, and VVX 600 series phones.

Note the following when using shared call appearance with ALU CTS:

● Members of the SCA group cannot resume remotely held calls.

● 21 Shared Call Appearances per line.

● The maximum number of calls associated with a shared call appearance group is the same as the number of calls provisioned for that shared line.

● An incoming call to a shared call appearance group can be presented to the group only as long as there is one available idle call appearance.

● All shared call appearances are able to receive and originate calls, regardless of the call activity on the other shared call appearances.

● Users can bridge into an active SCA call that is in shared mode.

Configure Shared Call Appearance

Parameter Function	template > parameter
Controls the default behavior of a Shared Call Appearance call. By default, an outgoing call from the call group is private. After the call is answered, the user needs to press the **Share** soft key to make the call public so that other people on the line can bridge in to the call.	**features.cfg** > feature.scap.defCallTypeExclusive
Specify the Hold request for Shared Call Appearance calls to the ALU server. This value must match what is configured on the ALU server for the Shared Call Appearance hold request.	**features.cfg** > feature.scap.HoldRequestUriUserPart

Visitor Desk Phone (VDP)

Visitor desk phone enables users registered with the Alcatel-Lucent CTS to access personal settings on a shared phone by logging in. Administrators configure a common setting for all phones and any user can make calls, including emergency calls, from a phone without having to log in. After the user logs in to the shared phone, personal settings are available as a user profile in <user> phones.cfg and any changes the user makes to phone settings are stored to this file. After the user logs out, another user can log into the phone to access their personal settings. Note that rebooting a phone logs out the user - the phone reboots with default settings and the user must log in. VDP is available on VVX 101, 201, 300 series, 400 series, 500 series, and 600 series business media phones.

On the server, you can configure the duration of a login period after which the user must re-enter credentials to the phone. When the time is nearing expiration, the server calls the phone and plays a message indicating the remaining time and prompts the user to re-enter credentials to extend the session.

If a user logs into a second phone when already logged into a first phone, the user is automatically logged out of the first phone. When logged in or out, users can dial an access code to play a message indicating if that user is logged in to a phone and the remaining time in a session.

The file <user>-directory.xml contains the user's contact list; the phone displays directory updates to the user at each login. Calls a user makes when logged into a phone are stored in call logs <user>-calls.xml. Calls a user makes when not logged in are not stored.

Note that any user phone services you enable, such as message-waiting indicator (MWI), busy lamp field (BLF), or shared call appearance (SCAP), are available to the user only after logging into the phone and the user profile is downloaded to the phone.

Configure VDP

Parameter Function	template > parameter
Enable or disable VDP and the Visitor login soft key.	**features.cfg** > feature.VDP.enabled
Specify the VDP login service access code.	**features.cfg** > prov.vdp.accessCode.login
Specify the VDP logout service access code.	**features.cfg** > prov.vdp.accessCode.logout

Configure Polycom Phones with GENBAND Server

GENBAND's application server, also called EXPERiUS™ A2, provides full-featured, IP-based multimedia communications applications for business and consumers. You can deploy EXPERiUS A2 as a standalone server or in combination with a GENBAND CONTiNUUM™ C20 server; features vary depending on your deployment. Polycom has performed interoperability tests with GENBAND C20 with Polycom VVX 300 series, 400 series, 500 series, 600 series, and 1500 phones.

UC Software 5.2 enhances Polycom's interoperability with GENBAND by offering several features on VVX business media phones while improving feature performance. Polycom UC Software 5.2.0 provides phones registered with GENBAND server the following new features or enhancements:

- MADN-SCA—Provides support for conference barge in, privacy, and remote call appearance. MADN-SCA requires you to deploy EXPERiUS A2 and CONTiNUUM C20 server.

- Global Address Book—The global address book (GAB) feature is a corporate directory application managed by the GENBAND server.

- Personal Address Book—The personal address book (PAB) feature is managed by the GENBAND server and allows multiple clients (phones, computer software) to read and modify a user's personal directory of contacts. When one client changes a contact all other clients are immediately notified of the change by the GENBAND server.

- E.911—Enhanced 911 services specific to GENBAND server implementation. You can use E.911 with a C20 server.

Configure Multiple Appearance Directory Number – Single Call Appearance (MADN-SCA)

Multiple appearance directory number—single call appearance (MADN-SCA) enables a group of users to share a single directory number that displays as a single line to each member of the group. When this feature is enabled, users can initiate or receive calls on this shared line. MADN-SCA requires you to deploy EXPERiUS A2 and CONTiNUUM C20 server.

When you set the line to shared, an incoming call alerts all the members of the group simultaneously, and the call can be answered by any group member. On the server, you can configure a privacy setting that determines whether or not, after the call is answered, other members of the group can barge in to the same call using the Barge In soft key and whether or not a call on hold can be picked up by other members of the group. Only one call can be active on the line at a time on the MADN-SCA shared line. When a call is in progress, any incoming calls to the line receive a busy tone.

Optionally, you can configure star codes on the server that you can dial on the phone to toggle the privacy setting during a single active call. Note the following call behavior. If the line is configured for privacy by default, you can use a star code to toggle privacy on and off during an active call. When the call ends, the line resets to privacy settings. If the line is configured on the server with privacy off, you can use a star code

to toggle to privacy on during an active call but you cannot toggle back to privacy off during the call. When the call ends, the line resets to privacy off.

In the UC Software download, Polycom provides the following two sample enhanced feature key (EFK) macros that you can configure to display on the phone to change privacy states: `privacyReleaseRestoreESK.cfg` and `privacyEnableESK.cfg`.

The following table lists all parameters available for MADN-SCA; see Example MDN-SCA Configuration for the minimum parameters you need to configure.

MADN-SCA Parameters

Parameter Function	template > parameter
Specify the user of the registration SIP URI.	**reg-basic.cfg** > reg.x.address
Per-registration parameter enabling the GENBAND MADN-SCA feature.	**reg-advanced.cfg** > reg.x.server.y.specialInterop
Global parameter enabling the GENBAND MADN-SCA feature.	**sip-interop.cfg** > voIpProt.server.x.specialInterop
Specify the line type as private or shared.	**reg-advanced.cfg** > reg.x.type
Enable or disable line barge-in.	**reg-advanced.cfg** > reg.x.bargeInEnabled
Set the maximum number of concurrent calls for a single registration.	**reg-advanced.cfg** > reg.x.callsPerLineKey
User ID to be used for authentication challenges for this registration.	**reg-basic.cfg** > reg.x.auth.userId
The password to be used for authentication challenges for this registration.	**reg-basic.cfg** > reg.x.auth.password
The IP address or hostname of the SIP server.	**reg-basic.cfg** > reg.x.outboundProxy.address
The domain of the authorization server.	**reg-advanced.cfg** > reg.x.auth.domain
This field must match the value of the registration which makes up the part of the shared line appearance.	**reg-advanced.cfg** > reg.x.thirdPartyName

Note: Setting up the MADN-SCA feature with GENBAND

If you configure the line-specific parameter **reg.x.server.y.address**, you must also configure values in the line-specific parameter r**eg.x.server.y.specialInterop**.

If you configure the global parameter **voIpProt.server.x.address**, you must also configure values in the global parameter **voIpProt.server.x.specialInterop**.

 For all deployments, including GENBAND, line-specific configuration parameters override global configuration parameters. If you set values in both line-specific and global parameters, line-specific parameters are applied and global parameters are not applied.

Example MDN-SCA Configuration

The following example configuration shows the minimum configuration you need to enable MDN-SCA on the phone. You can use the parameters in the template configuration files or create your own configuration file from the parameters.

To configure MADN-SCA:

1   Enter values for the following parameters in a configuration file and save. The value `8630@polycom.com` is an example registration address.
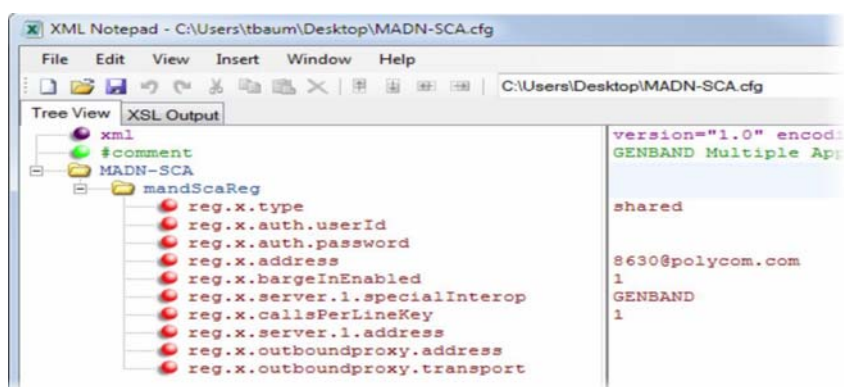
2   Enter the name of the configuration file to the CONFIG_FILES field of the master configuration file and save.

Configure the Global Address Book (GAB)

GENBAND's global address book (GAB) is a read-only global contact directory set up by an administrator and can co-exist with other corporate directories on the phone. Users can access the GAB on the phone in the Directories menu and Features menu.

GAB Parameters

Parameter Function	**template** > parameter
Enable or disable the global address book.	**features.cfg** > feature.corporateDirectory.alt.enabled
Enter the URL address of the GAB service provided by the server.	**features.cfg** > dir.corp.alt.address
The port that connects to the server if a full URL is not provided.	**features.cfg** > dir.corp.alt.port
The user name used to authenticate to the GENBAND server.	**features.cfg** > dir.corp.alt.user
Displays the results from your last address directory search.	**features.cfg** > dir.corp.alt.viewPersistence
Use a filter to set a predefined search string through configuration files.	**features.cfg** > dir.corp.alt.attribute.x.filter

GAB Parameters

Enable or disable a filter string.	**features.cfg** > dir.corp.alt.attribute.x.sticky
Provide a label to identify a user.	**features.cfg** > dir.corp.alt.attribute.x.label
The name of the parameter to match on the server.	**features.cfg** > dir.corp.alt.attribute.x.name
Define how parameter x is interpreted by the phone.	**features.cfg** > dir.corp.alt.attribute.x.type
Specify a method for synchronizing the directory and server.	**sip-interop.cfg**, **site.cfg**, > dir.local.serverFeatureControl.method

Example GAB Configuration

The following example shows the minimum parameters you need to configure to enable GAB on the phone.

To configure GAB:

1   Enable GAB by configuring the values in `feature.corporateDirectory.alt` and `dir.corp.alt`. The following illustration includes an example GAB address book parameters in `dir.corp.alt.attribute`.
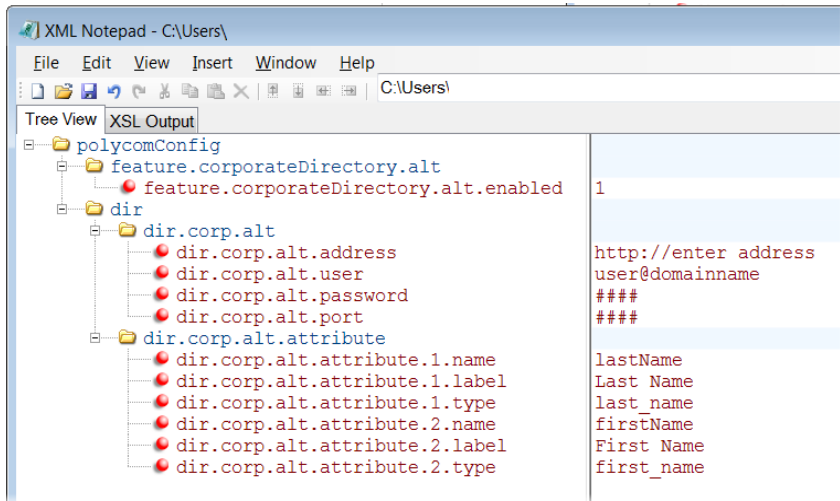
2   Save the configuration file.

3   Enter the name of the configuration file to the CONFIG_FILES field of the master configuration file and save.

Configure the Personal Address Book (PAB)

The personal address book (PAB) enables users to read and modify a personal directory of contacts on their phone. Users can access the PAB on the phone in the Features or Directory menu.

When you modify contact information using any soft client, desk phone, or mobile client registered to the same line, the change is made on all other clients, and you are notified immediately of the change by the GENBAND server. When you enable server control, five telephone number fields per contact are available.

PAB Parameters

Parameter Function	template > parameter
Enable or disable the global address book.	**features.cfg** > feature.corporateDirectory.alt.enabled
Enter the URL address of the GAB service provided by the server.	**features.cfg** > dir.corp.alt.address
The port that connects to the server if a full URL is not provided.	**features.cfg** > dir.corp.alt.port
The user name used to authenticate to the GENBAND server.	**features.cfg** > dir.corp.alt.user
Displays the results from your last address directory search.	**features.cfg** > dir.corp.alt.viewPersistence
Use a filter to set a predefined search string through configuration files.	**features.cfg** > dir.corp.alt.attribute.x.filter
Enable or disable a filter string.	**features.cfg** > dir.corp.alt.attribute.x.sticky
Provide a label to identify a user.	**features.cfg** > dir.corp.alt.attribute.x.label
The name of the parameter to match on the server.	**features.cfg** > dir.corp.alt.attribute.x.name
Define how parameter x is interpreted by the phone.	**features.cfg** > dir.corp.alt.attribute.x.type
Specify a method for synchronizing the directory and server.	**sip-interop.cfg**, **site.cfg**, > dir.local.serverFeatureControl.method
Specify the phone line to enable personnel address book feature on.	**site.cfg** > dir.local.serverFeatureControl.reg
Specify the maximum number of contacts available in the GENBAND personnel address book contact directory.	**site.cfg** > dir.genband.local.contacts.maxSize

Example PAB Configuration

The following example shows an example PAB configuration.

To configure PAB:

1 Enter the values shown for the following parameters and save the configuration file.
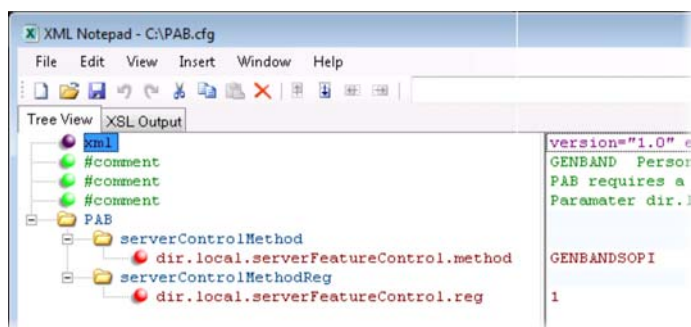
2 Enter the configuration file to the CONFIG_FILES field of the master configuration file and save.

Configure GENBAND Enhanced 911 (E.911) Location

When using E.911 with GENBAND, you must correctly configure a location tree on the server that the phone downloads on start up. When the phone starts up, the phone prompts you to choose a location, which is stored on the phone to identify the phone location to 911 operators dispatching emergency services. This feature is available for all VVX phones and is disabled by default as it is available only in a GENBAND environment.

Callers can make a 911 call when the phone is locked, regardless of the call state, or when other features are in use. When a 911 call is in progress, the interface soft key options to end, hold, or transfer a call do not display, you cannot use hard keys on the VVX 300/301, 310/311, 400/401, 410/411, and 1500 to end, hold, or transfer a call, and DND or call forwarding enabled do not work.

To set the location information on the phone:

1   Register the phone.

   The phone prompts you with a warning message 'Set your location' for 10 seconds.

2   Press the warning message to enter a location. If the warning message disappears, on the phone, go to **Settings > Status > Diagnostics > Warnings**.

   The warning message 'Set your Location' displays until you press Clear.

3   Press the **Details** soft key to enter a location to the location tree navigation menu. Example navigation menus are shown in the following figures.

4   Choose a location and press **Save**.

5   On the phone, go to **Status > Location Information**.

   The location information displays in the **Status** menu.

The following figure shows the menu structure for setting the Location Status on VVX phones.

Location menu structure for VVX 500/501, 600/601, 1500

E911 Parameters

Parameter Function	**template** > parameter
Enable or disable the GENBAND E.911 feature.	**features.cfg** > feature.genband.E911.enabled
Enter a description of the location of the phone.	**site.cfg** > genband.E911.location.description

E911 Parameters

Enter the location ID corresponding to the location description you entered in `genband.E911.location.description`.	**site.cfg** > genband.E911.location.locationID
Select the registration line to use to retrieve E.911 location information	**reg-basic.cfg** > genband.E911.registration.line

Configure Polycom Phones with BroadSoft

This section shows you how to configure Polycom devices with BroadSoft Server options. You can use the features available on the BroadWorks R18 server or the BroadWorks R20 or later server with the following phones: VVX 300 series, 400 series, 500 series, 600 series, and 1500 phones. Note that you cannot register lines with the BroadWorks R18 server and the R20 and later server on the same phone. All lines on the phone must be registered to the same BroadWorks server.

Some features require you to authenticate the phone with the BroadWorks XSP service interface as described above in the section Configure Authentication for BroadSoft BroadWorks Xtended Service Platform (XSP) Service Interface.

Support for BroadWorks Call Decline Policy

For shared lines in a BroadSoft BroadWorks environment, you can set the parameter `call.shared.reject` to enable users to reject calls on the shared line. When a user rejects a call to the shared line, the call is rejected on all phones registered with the shared line.

BroadWorks Shared Lines

Parameter Template	Permitted Values
`call.shared.reject` sip-interop.cfg	For shared line calls on the BroadWorks server. 0 - The phone displays a Reject soft key to reject an incoming call to a shared line. 1 - The Reject soft key does not display.

BroadSoft Flexible Seating

Available on the BroadWorks R20 server, Flexible Seating allows you to configure a host phone that users can log in to and access their registered line. After the user logs in, user settings are replicated on the host phone as a guest profile and the registered phone line is then active on both the primary phone and the host phone.

You have the option to configure a PIN for the guest profile. Users can access their guest profile on a host phone using a PIN which prevents others from logging them out with the phone password. The PIN takes precedence over the local phone password and the guest user must log out of the phone with the PIN before another user can log in with their password. On the BroadWorks server, you can configure a period of time after which the server automatically logs out a user from a phone in case a user does not log out.

You can configure each Guest Profile to replicate the BroadSoft UC-One contact directory and favorites to the host phone, but the Polycom contact directory and favorites cannot be replicated. Once a user is logged in to a host phone, any settings changes are not automatically synchronized between the host phone and the primary phone. Settings can be manually synchronized from the BroadSoft BroadWorks server.

If no user is logged in, the phone cannot be used to accept incoming calls. If a user is logged in and Phone Lock is enabled, the phone cannot be used to call numbers on the authorized call list. In either state, the phone can be used to place outgoing calls to emergency numbers that you define on the BroadWorks server.

Administrators can change the host phone's user password from the Web Configuration Utility at any time. You can also change the password from the phone screen when the guest user is not logged in. Guest users cannot change the host phone's user password. The following features are not supported on phones configured for Flexible Seating:

● Hoteling

● Visitor Desk Phone (VDP)

● User Profile Feature

The following features are supported and available to users on the phone when configured on the server and are not supported if configured locally:

● Call Forwarding

● Do Not Disturb (DND)

Configure Flexible Seating

To configure a host phone, create two profiles on the BroadSoft BroadWorks server: a host user profile and a guest user profile. The host user account is a placeholder account that supports guest users. In the host user profile configuration files, add the configuration parameters shown in the following table and map these parameters to the corresponding BroadSoft BroadWorks configuration tags.

Polycom Configuration Parameter	BroadSoft Tag
`hoteling.reg`	%BWHOTELINGLINE-x%
`hotelingMode.type`	%BWHOTELINGMODE-x%
`fs.unLockPhone.pin`	%BWFLEXIBLESEATINGUNLOCKPIN-x%

Flexible Seating Parameters

Parameter Template	Permitted Values
`fs.unLockPhone.pin`	NULL (default) 4 - 10 digits Set a security pin for the Flexible Seating guest line on the host phone.
`hoteling.reg` features.cfg	1 (default) - Specifies the phone line on the host phone which hosts the guest line.
`hotelingMode.type`	-1 (Default): The parameter does not exist on the Broadsoft server. 0 - Both Flexible Seating and Hoteling are disabled on the BroadSoft Device Management Server (DMS). 1 - Hoteling is enabled 2 - Flexible Seating is enabled but guest is not logged in. 3 - Flexible seating location is enabled and guest is logged in

BroadSoft Executive-Assistant

Using configuration files, you can enable the BroadSoft Executive-Assistant feature on lines registered with the BroadWorks R20 or later server, and assign lines as an executive or an assistant. Note that all corresponding Executive and Assistant lines must be registered to the same server.

After you enable the feature, users can set call filters that determine whether calls go to the assistant or to the executive first. Executives can set filters for themselves and assistants can set filters for the executive. Executives and assistants can also enable screening, which enables the executive's phone to display the incoming call notification for all filtered calls.

To use this feature on Polycom phones registered with a BroadWorks server, you must authenticate the phone with the BroadSoft XSP service interface.

Configure BroadSoft Executive-Assistant

In the BroadWorks Web Portal, you must enable the Executive Service for private and shared executive lines, and the Executive-Assistant Service for private and shared assistant lines.

The BroadWorks server allows the following configuration options: Executive private line, Executive-Assistant Service line, and a shared alias line. Administrators can set up executive and assistant lines in the following scenarios:

- A private executive line with an assistant with a private line
- Shared executive line with an assistant with a private line
- Shared executive line with a shared line alias on the assistant's phone
    - The shared line must be created as a shared location of a line with the Executive Service on the BroadWorks server.
    - In this option, the main line registration is a private line for the assistant, and the secondary registration is a shared line for the executive.

The following table includes the configuration parameters you can use to enable and configure the Executive-Assistant feature.

BroadSoft Executive-Assistant Configuration Parameters

Parameter Template	Permitted Values
`feature.BSExecutiveAssistant.enabled` features.cfg	0 (default) - Disables the BroadSoft Executive-Assistant feature. 1 - Enables the BroadSoft Executive-Assistant feature.

BroadSoft Executive-Assistant Configuration Parameters

Parameter Template	Permitted Values
`feature.BSExecutiveAssistant.regIndex` features.cfg	The registered line assigned to the executive or assistant for the BroadSoft Executive-Assistant feature. 1 (default) to 255 - The registered line for the Executive or Assistant. Note that a line icon for the role specified by the parameter `feature.BSExecutiveAssistant.userRole` displays even if you do not assign an Executive-Assistant service to a line in the BroadSoft Web Portal. Ensure that the services assigned to the line match the user role.
`feature.BSExecutiveAssistant.userRole` features.cfg	ExecutiveRole (default) - Sets the registered line as an Executive line. AssistantRole - Sets the registered line as an Assistant line. Note: A phone can have a line set as an Executive or an Assistant; an Executive and an Assistant line cannot be set on the same phone.

Configure Authentication for BroadWorks Xtended Service Platform (XSP) Service Interface

You can configure Polycom VVX business media phones to use advanced features available with BroadSoft BroadWorks server. The VVX business media phones support the following he advanced BroadSoft features:

● BroadSoft Enhanced Call Park

● BroadSoft UC-One directory, favorites, and presence

● BroadSoft UC-One personal call control features

To use these features on Polycom devices with a BroadWorks server, you must authenticate the phone with the BroadSoft XSP service interface. The authentication method to use depends on which version of BroadWorks you are running. If your server is running BroadWorks R19 or earlier, enable the following parameters to authenticate on the BroadWorks server using separate XSP credentials:

● `dir.broadsoft.xsp.address`

● `reg.x.broadsoft.userId`

● `reg.x.broadsoft.xsp.password`

● `reg.x.broadsoft.useXspCredentials`

If your server is running BroadWorks R19 Service Pack 1 or later, enable the following parameters to authenticate on the BroadWorks server using the same SIP credentials you used to register the phone lines: dir.broadsoft.xsp.address

● `reg.x.auth.userId`

● `reg.x.auth.password`

● `reg.x.broadsoft.userId`

Configure BroadWorks XSP Service Interface Authentication

Parameter Function	**template** > parameter
Enter the password associated with the BroadSoft XSP user account for the line. Required only when `reg.x.broadsoft.useXspCredentials=1.`	**features.cfg** > reg.x.broadsoft.xsp.password
The BroadSoft user ID to authenticate with the BroadSoft XSP service interface, If this parameter value is empty, the line might be registered with BroadWorks server, and not have access to advanced features.	**features.cfg** > reg.x.broadsoft.userId
Determine the XSP authentication method for the BroadWorks version you are using.	**features.cfg** > reg.x.broadsoft.useXspCredentials
Set the BroadSoft Directory XSP address.	**features.cfg** > dir.broadsoft.xsp.address
User ID to be used for authentication challenges for this registration when `reg.x.broadsoft.useXspCredentials=0.`	**reg-basic.cfg** > reg.x.auth.userId
The password to be used for authentication challenges for this registration when `reg.x.broadsoft.useXspCredentials=0.`	**reg-basic.cfg** > reg.x.auth.password

BroadWorks Enhanced Call Park

You can configure BroadWorks Enhanced Call Park per registered line. The following features are available for Enhanced Call Park:

- You can configure Enhanced Call Park only using configuration files; you cannot configure the feature on the Web Configuration Utility or from the local phone interface.
- You can configure Enhanced Call Park for private lines and shared lines. No configuration is necessary to enable the call park notification for monitored BLF lines.
- The default star codes set for the `call.parkedCallRetrieveString` is *88.

The parameter `call.parkedCallRetrieveString` is updated for this feature to configure the access code to retrieve calls.

Configure Enhanced Call Park

Parameter Function	**template** > parameter
Enable or disable the BroadWorks Enhanced Call Park feature.	**reg-advanced.cfg** > reg.x.enhancedCallPark.enabled
The line extension for a shared line. If there is no extension provided for this parameter, the call park notification is ignored for the shared line.	**reg-advanced.cfg** > reg.x.lineAddress

Configure Enhanced Call Park

Enable or disable the audio notifications for parked calls on private and shared lines.	**features.cfg** > feature.enhancedCallPark.allowAudioNotification
The access code used to retrieve a parked call.	**features.cfg** > call.parkedCallRetrieveString

Configure Polycom BroadSoft UC-One Application

The Polycom BroadSoft UC-One application integrates with BroadSoft Enterprise Directory and BroadCloud services—a set of hosted services by BroadSoft—to provide the following three features:

● **BroadSoft Directory**   Displays information for all users in the enterprise, for example, work and mobile phone numbers.

● **BroadCloud Presence**   Enables users to share presence information with the BroadTouch Business Communicator (BTBC) client application.

● **BroadCloud Favorites**   Enables users to mark contacts as favorites with the BroadTouch Business Communicator (BTBC) client application.

These features are available on Polycom VVX 500 series and VVX 600 series business media phones running Polycom UC Software 4.1.3G or later, and VVX 300 series and VVX 400 series business media phones running Polycom UC Software 5.0.0 or later. These features require support from the BroadSoft BroadWorks R18 SP1 platform with patches and BroadSoft BroadCloud services. For details on how to set up and use these features, see the latest *Polycom VVX Business Media Phones - User Guide* at Latest Polycom UC Software Release.

Polycom's BroadSoft UC-One application enables you to:

● Access the BroadSoft Directory

● Search for contacts in BroadSoft Directory

● View BroadSoft UC-One contacts and groups

● View the presence status of BroadSoft UC-One contacts

● View and filter BroadSoft UC-One contacts

● Activate and control BroadSoft UC-One personal call control features.

The table Configure the BroadSoft UC-One Application lists all parameters available that configure features in the BroadSoft UC-One application.

Administrators can configure the UC-One Call Settings menu and menu options in the Web Configuration Utility.

To enable UC-One Call Settings in the Web Configuration Utility:

1 Navigate to **Settings > UC-One**.

2 Under **General**, click **Enable** for **BroadSoft UC-One**.

This enables the UC-One Call Settings menu to display on the phone.

To enable UC-One Call Settings menu options:

1 In the Web Configuration Utility, navigate to **Settings > UC-One**.

2 Under **Call Settings Features**, enable each feature menu you want available on the phone.

The following table lists all parameters available to configure features in the BroadSoft UC-One application.

Configure the BroadSoft UC-One Application

Parameter Function	template > parameter
To turn QML application on or off and enable or disable display of the user interface for BroadSoft UC-One directory.	**features.cfg** > feature.qml.enabled
To turn BroadSoft Directory on or off.	**features.cfg** > feature.broadsoftdir.enabled
To turn BroadSoft UC-One on or off.	**features.cfg** > feature.broadsoftUcOne.enabled
To turn Presence on or off.	**features.cfg** > feature.presence.enabled
Enable or disable the UC-One Settings icon to display on the Home screen.	**features.cfg** > homeScreen.UCOne.enable
To set the BroadSoft Directory XSP home address.	**applications.cfg** > dir.broadsoft.xsp.address
To set the BroadSoft Directory XSP user name.	**applications.cfg** > dir.broadsoft.xsp.username
To set the BroadSoft Directory XSP password.	**applications.cfg** > dir.broadsoft.xsp.password
To set the BroadSoft XMPP password.	**features.cfg** > xmpp.1.auth.password
To set the BroadSoft XMPP dial method.	**features.cfg** > xmpp.1.dialMethod
To turn BroadSoft XMPP presence on or off.	**features.cfg** > xmpp.1.enable
To set the BroadSoft XMPP Jabber Identity used to register with presence server.	**features.cfg** > xmpp.1.jid
To turn the BroadSoft XMPP inviter's subscription for presence.	**features.cfg** > xmpp.1.roster.invite.accept
To set the BroadSoft XMPP presence server IP or FQDN.	**features.cfg** > xmpp.1.server
To turn the verification of the TLS certificate provided by the BroadSoft XMPP presence server on or off.	**features.cfg** > xmpp.1.verifyCert

Example BroadSoft UC-One Configuration

The illustration shown next provides an example configuration for the BroadSoft UC-One features.
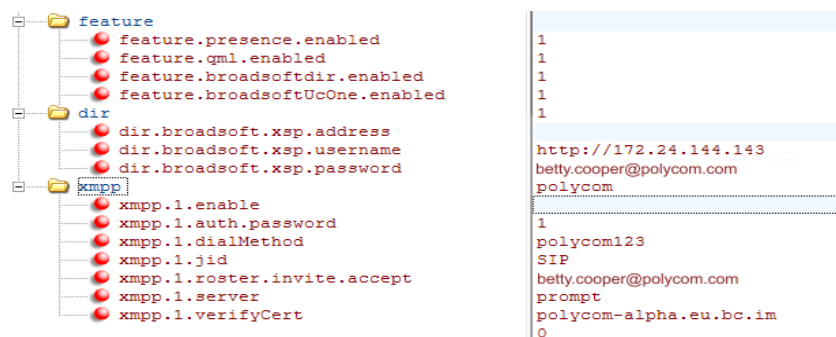
Anonymous Call Rejection

Anonymous Call Rejection enables users to automatically reject incoming calls from anonymous parties who have restricted their caller identification. You can enable the Anonymous Call Rejection feature using configuration files or the Web Configuration Utility. After you enable the feature for users, users can turn call rejection on or off from the phone. When a user turns Anonymous Call Rejection on, the phone gives no indication that an anonymous call was received.

You can configure this option in the Web Configuration Utility.

To enable Anonymous Call Rejection in the Web Configuration Utility:

1   Navigate to **Settings > UC-One**.
2   Under the **Call Setting Features**, click **Enable** for **Anonymous Call Rejection**.

Configure Anonymous Call Rejection

Parameter Function	template > parameter
Displays the Anonymous Call Rejection menu on the phone.	**features.cfg** > feature.broadsoft.xsi.AnonymousCalReject.enabled
Enable or disable all BroadSoft UC-One features.	**features.cfg** > feature.broadsoftUcOne.enabled
Enter the BroadSoft user ID to confirm that the line is a registered BroadSoft line.	**features.cfg** > reg.x.broadsoft.userId

Simultaneous Ring Personal

The Simultaneous Ring feature enables users to add phone numbers to a list of contacts whose phones ring simultaneously when the user receives an incoming call. When you enable the Simultaneous Ring menu option on the phones for users, users can turn the feature on or off from the phone and define which numbers should be included in the Simultaneous Ring group.

You can enable or disable the Simultaneous Ring feature for users using configuration files or the Web Configuration Utility.

Configure Simultaneous Ring

Parameter Function	template > parameter
Enable or disable the Simultaneous Ring Personal feature.	**features.cfg** > feature.broadsoft.xsi.SimultaneousRing.enabled
Enable or disable all BroadSoft UC-One features.	**features.cfg** > feature.broadsoftUcOne.enabled

Line ID Blocking

You can enable or disable the Line ID Blocking menu option on the phone. When you enable the menu for users, users can choose to hide their phone number before making a call. You can configure this feature using configuration parameters or the Web Configuration Utility.

Configure Line ID Blocking

Parameter Function	template > parameter
Enable or disable the Line ID Blocking feature.	**features.cfg** > feature.broadsoft.xsi.LineIdblock.enabled
Enable or disable all BroadSoft UC-One features. If disabled, none of the UC-One features are enabled even if the individual feature parameters are enabled.	**features.cfg** > feature.broadsoftUcOne.enabled

BroadWorks Anywhere

BroadWorks Anywhere enables users to use one phone number to receive calls to and dial out from their desk phone, mobile phone, or home office phone. When you enable this feature, users can move calls between phones and perform phone functions from any phone. When enabled, the BroadWorks Anywhere settings menu displays on the phone and users can turn the feature on or off and add BroadWorks Anywhere locations on the phone. You can configure BroadWorks Anywhere using configuration files or the Web Configuration Utility.

Configure BroadWorks Anywhere

Parameter Function	template > parameter
Enable or disable the BroadWorks Anywhere feature. If set to 0, the feature menu is disabled does not display.	**features.cfg** > feature.broadsoft.xsi.BroadWorksAnywhere.enabled
Enable or disable all BroadSoft UC-One features. If disabled, none of the UC-One features are enabled even if the individual feature parameters are enabled.	**features.cfg** > feature.broadsoftUcOne.enabled

Configure Remote Office

Remote Office enables users to set up a phone number on their office phone to forward incoming calls to a mobile device or home office number. When enabled, this feature enables users to answer incoming calls to the office phone on the phone, and any calls placed from that phone show the office phone number. You can configure Remote Office using configuration files or the Web Configuration Utility.

Configure Remote Office

Parameter Function	template > parameter
Enable or disable the Remote Office feature.	**reg-advanced.cfg** > feature.broadsoft.xsi.RemoteOffice.enabled
Enter the BroadSoft user ID to confirm that the line is a registered BroadSoft line. If empty, the line is not considered as a BroadSoft line.	**features.cfg** > reg.x.broadsoft.userId

Configure Remote Office

Enable or disable all BroadSoft UC-One features. If disabled, none of the UC-One features are enabled even if the individual feature parameters are enabled.	**features.cfg** > feature.broadsoftUcOne.enabled
Set the BroadSoft Directory XSP password.	**applications.cfg** > dir.broadsoft.xsp.password

BroadSoft UC-One Credentials

Enabling this feature allows users to enter their BroadWorks UC-One credentials on the phone instead of in the configuration files. The parameters `reg.x.broadsoft.useXspCredentials`, and `feature.broadsoftUcOne.enabled` must be enabled to display the UC-One Credentials menu option on the phone.

Configure XSP User Name an Password

Parameter Function	template > parameter
Set the IP address or hostname of the BroadSoft directory XSP home address.	**features.cfg** > dir.broadsoft.xsp.address
Enter the BroadSoft user ID to confirm that the line is a registered BroadSoft line. If empty, the line is not considered as a BroadSoft line.	**features.cfg** > reg.x.broadsoft.userId
Enable or disable all BroadSoft UC-One features. If disabled, none of the UC-One features are enabled even if the individual feature parameters are enabled.	**features.cfg** > feature.broadsoftUcOne.enabled
Set the BroadSoft Directory XSP user name. Note that this value will be overridden by what the user enters through the phone UI	**applications.cfg** > dir.broadsoft.xsp.username
Set the BroadSoft Directory XSP password. Note that this value will be overridden by what the user enters through the phone UI	**applications.cfg** > dir.broadsoft.xsp.password
Turn BroadSoft Directory on or off.	**features.cfg** > feature.broadsoftdir.enabled

Set Up Polycom Phones for Use with Microsoft

Skype for Business and Lync Server provides 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.Note that the features available when you are registered with Skype for Business Server vary with the Polycom phone model and Polycom UC Software version you are using. Note that UC Software 5.3.0 does not support the use of VVX 1500 business media phone with Skype for Business Server. Note that the concurrent failover/fallback feature is not compatible in a Microsoft environment.

You can deploy Polycom devices with Microsoft® Skype™ for Business Online, Microsoft® Skype™ for Business 2015, Microsoft® Lync® 2013, or Microsoft® Lync® 2010 on-premises. As of UC Software 5.4.0A, you can register Polycom devices with Microsoft Skype for Business Online.

As of UC Software 5.1.2, Polycom offers devices with an Open SIP or a Skype for Business base profile (a Lync SKU). Polycom devices shipped with a Lync base profile include Skype for Business-qualified UC Software with a feature license included and enable you to start up the phone and register with Skype for Business Server with default settings.

If you are using UC Software 5.3.0 with Skype for Business Server and want to change default settings or customize your deployment, you must set up a provisioning server.

For a list of available features and instructions on deploying RealPresence Trio 8800 solution with Skype for Business and Lync Server, see the latest *Polycom UC Software in a Microsoft Environment – Deployment Guide* at RealPresence Trio on Polycom Support.

Polycom UC Software 5.4.1 supports the following devices with Lync Server and Skype for Business:

- VVX 201, 300 series, 400 series, 500 series, and 600seriesbusiness media phones
- SoundStructure VoIP Interface

Polycom UC Software 5.4.0A supports the following devices with Lync Server and Skype for Business:

- VVX 201, 300, 310, 400, 410, 500, and 600 business media phones
- SoundStructure VoIP Interface

You can register the following phones with Lync Server 2010 and Lync Server 2013 using UC Software 4.1.x and later:

- VVX 300, 310, 400, 410, 500, and 600
- SoundStructure VoIP Interface

Note the following points when using Polycom phones with Skype for Business Server:

- Polycom UC Software enables you to register a single phone line with Skype for Business Server; you cannot register multiple lines with Skype for Business Server. When you register a line on a Polycom phone using Skype for Business Server you cannot register lines with another server.

Polycom phones shipped with a Lync Base Profile enabled (Skype for Business, Lync Server, and O365 SKU) include the required Polycom UC Software license. If you are not using Polycom phones shipped with a Lync Base Profile, you must purchase a Skype for Business Feature License from a Polycom reseller or Polycom sales representative to use Polycom VVX products in a Microsoft Skype for Business environment. You can use Polycom phones in a Skype for Business, Lync Server, or O365 environment for trial purposes, without purchasing a license, to a maximum of 30 days.

You must purchase a Polycom UC Software license for a RealPresence Trio system that ship without the Lync Base Profile enabled to use RealPresence Trio systems in a Skype for Business, Lync Server, and O365 environment.

- When you are running UC Software 4.1.x or 5.0 for use with Lync Server 2010, you have access to two separate contact lists: the default local contact directory on your Polycom phone and a Lync contact list. If you want to disable the local contact directory on your Polycom phone or make it read-only, see the section Use the Local Contact Directory.

Tip: Workaround for phones using G.722 and retrieving Microsoft Skype for Business voicemail

If your Polycom phones are configured with G.722 and users find that they do not hear audio when retrieving voicemail from the Microsoft Skype for Business 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

As of UC Software 4.0.1, VVX phones 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 widget that displays in the status bar on the VVX phone.

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 **Menu > Settings > Basic > Login Credentials**.

You can view the calendar information in day or month format. On VVX phones, the meeting details displays beside the calendar view.

All possible phone numbers that you can dial to place a call to the meeting displays 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 plays 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 user guide for your phone at Polycom Voice Support.

Enable Microsoft Exchange Calendar Integration

Parameter Function	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

Example Exchange Calendar Configuration

The following example shows the Calendar feature enabled in features.cfg.
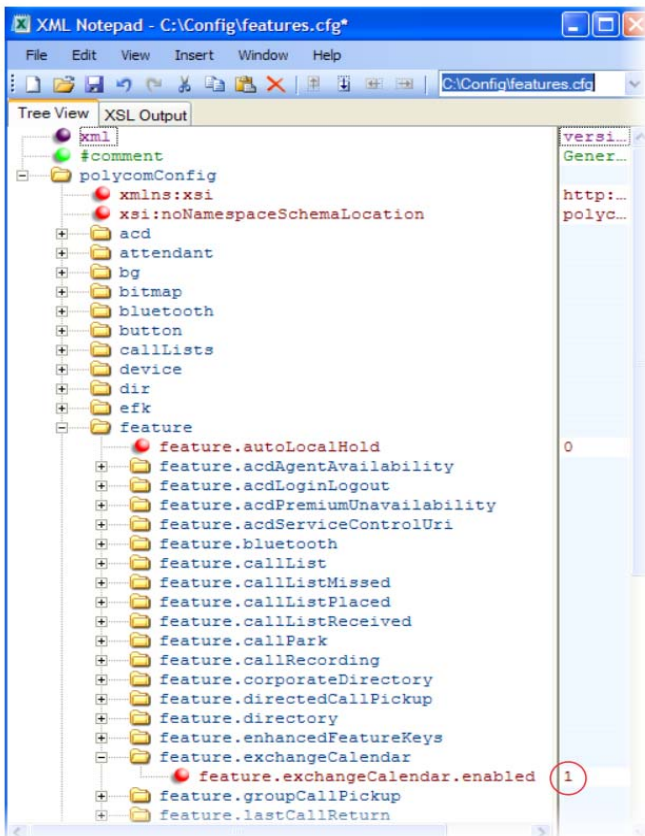
After you enable the feature, specify the Microsoft Exchange Server address in applications.cfg, as shown next. In this example, a pattern has been specified for meeting numbers. When you specify a pattern, any number in your meeting invitation that matches the pattern displays on a meeting participants' 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.
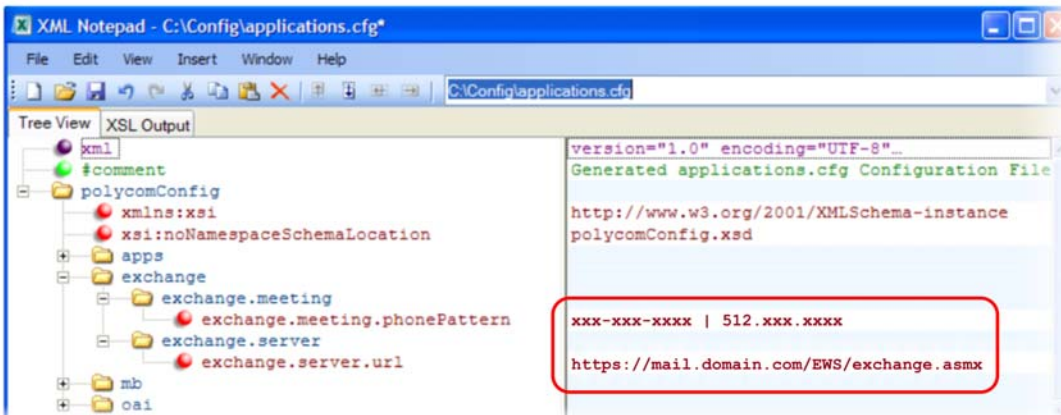
Configure Phone Hardware

This section provides information on configuring the phone hardware.

Connect to an Ethernet Switch

VVX business media phones have two Ethernet ports—labeled LAN and PC— and an embedded Ethernet switch that runs at full line rate. The SoundStructure VoIP Interface has one Ethernet port, labeled LAN. The Ethernet switch enables you to connect a personal computer and other Ethernet devices to the office LAN by daisy-chaining through the phone, eliminating the need for a stand-alone hub.

Each phone can be powered through an AC adapter or through a Power over Ethernet (PoE) cable connected to the phone's LAN port. To disable the PC Ethernet port, see the section Disable the PC Ethernet Port.

If you are using a VLAN, ensure that the 802.1p priorities for both default and real-time transport protocol (RTP) packet types are set to 2 or greater so that audio packets from the phone have priority over packets from the PC port. For more information, see <qos/>.

USB Port Lockdown

The USB Port Lock down feature enables you to choose which phone headset ports to enable or disable and choose which of the phone's USB ports to power on or off.

The port lockdown feature is available on the VVX 500/501, 600/601, and 1500 phones. The VVX 1500 has a single USB port, and the VVX 500/501 and 600/601 support two USB ports, one port on the top and rear of the phones. The phones support various USB devices such as USB mass storage devices and a USB headset. The top USB port on the VVX 500/501 and 600/601 support the VVX Camera.

> **Caution: Remove and replace feature.usbTop.power.enabled**
> Two parameters `feature.usbTop.power.enabled` and `feature.usbRear.power.enabled` replace `feature.usb.power.enabled`. You must replace feature.usb.power.enabled with these two new parameters and set both to 0 to disable USB ports.

Note how the parameters affect ports on the phones:

- The parameters `feature.usbTop.power.enabled` and `feature.usbRear.power.enabled` enable and disable the USB ports on VVX 401/411, 500/501, 600/601, and 1500 phones.

- Only the parameter `feature.usbTop.power.enabled` applies to VVX 401/411 and 1500 phones. The parameter `feature.usbRear.power.enabled` does not apply to VVX 401/411 and 1500 phones because the VVX 401/411 and 1500 only have one USB port.

- You can control the VVX 500/501 and 600/601 top and rear USB ports independently using `feature.usbTop.power.enabled` to control the top USB port and `feature.usbRear.power.enabled` to control the rear USB port.

> **Note: Disabling the top USB port on VVX 500/501 and 600/601 phones**
> If you set the parameter `feature.usbTop.power.enabled` to 0 to disable the top USB port on VVX 500/501 and 600/601 phones, you must set the parameter `video.enable` to 0 as well.

The top and rear USB ports are enabled by default. Disabling the USB port makes the following features unavailable:

- Call recording
- Picture frame
- USB headset
- USB camera for video calls on the VVX 500/501 and 600/601 - no video calls
- USB charging device on the rear port of the VVX 500/501 and 600/601

> **Note: A power adapter can cause issues on VVX 500/501 phones**
> When you connect a power adapter to a VVX 500/501, the USB ports are powered on even if the parameters feature.usbTop.power.enabled and feature.usbRear.power.enabled are disabled. This can cause issues during phone reboots when USB devices are connected to the phone.

Configure USB Port Lockdown

Parameter Function	template > parameter
Enable or disable power to USB ports.	**template** > feature.usbTop.power.enabled
Enable or disable power to USB port 2.	**template** > feature.usbRear.power.enabled
Enable or disable power to the top USB port on VVX 500/501 and 600/601 phones.	**video.cfg** > video.enable

Connect Polycom VVX Expansion Modules

The Polycom VVX Expansion Modules are consoles you can connect to Polycom VVX business media phones to add additional lines. VVX Expansion Modules enable you to handle large call volumes on a daily basis and expand the functions of your phone. You can accept, screen, dispatch, and monitor calls with VVX Expansion Module and reduce the number of lost customer calls, shorten transaction times, and increase the accuracy of call routing.

You can capture the current screen of expansion modules, see the section

Polycom VVX Expansion Modules are available for the following Polycom VVX business media phones running UC Software 4.1.6 or later:

- VVX 300 series and 400 series business media phones
- VVX 500 series and 600 series business media phones

> **Web Info: Polycom Support documentation for expansion modules**
> For all documents that help you set up and use the Polycom VVX expansion modules with your VVX phones see Polycom VVX Expansion Modules Support page.

The following features are available on the VVX LCD Color Expansion Modules and VVX Expansion Modules with a paper display:

- **VVX Expansion Modules – LCD Color Display**   VVX Color Expansion Modules feature an easy-to-navigate 272x480 LCD display. Each color expansion module provides you with 28 line keys and 3 display pages, supporting a total of 84 lines that you can set up as registrations, favorites, busy

lamp field contacts, or Microsoft Skype for Business presence contacts. You can connect up to three color expansion modules to your phone to support an additional 252 line keys per phone. If you are registering Polycom phones with Skype for Business Server, you can use only the LCD color display expansion modules; you cannot use the paper display expansion modules for phones registered with Skype for Business Server.

● **VVX Expansion Modules – Paper Display**   VVX paper display expansion modules provide you with 40 line keys that you can set up as registrations, favorites, or busy lamp field contacts. You can connect up to three expansion modules to your phone to support an additional 120 line keys per phone.

> **Note: Line keys are numbered sequentially**
>
> The number of line keys the phone supports varies by phone model. For a list of see the column in the table Flexible Call Appearances. Line keys on VVX phones and expansion modules are numbered sequentially, and the line key numbering on your expansion module depends on how many lines your phone supports. For example, a VVX 600/601 phone supports 16 lines, numbered 1-16. The first line on an expansion module connected to a VVX 600/601 phone is line 17.

The following figure illustrates the LCD color and paper expansion modules.

Polycom VVX Expansion Modules

Expansion Module LCD color display and paper display

VVX Expansion Module Power Values

Polycom VVX phones use more power when you connect expansion module. The following table outlines the power each phone uses when you connect an expansion module, as well as the power value sent in LLDP-MED. For a list of power values for all Polycom phones without an expansion module attached, see Power Values.

VVX Expansion Module Power Values

Model	Power Usage (Watts)	Power Value Sent in LLDP-MED Extended Power Via MDI TLV
VVX 300/301	5.0	5000mW
VVX 310/311	5.0	5000mW

VVX Expansion Module Power Values

VVX 400/401	5.0	5000mW
VVX 410/411	5.0	5000mW
VVX 500/501	8.0	8000mW
VVX 600/601	8.0	8000mW

Generate Configured Line Key Information

Using the Web Configuration Utility, you can generate and download a PDF file with the line key configuration for each paper display expansion module connected to your VVX phone. The generated PDF enables you to print line key information for line keys on your expansion modules and insert the PDF as a directory card on your modules.

To generate and download the line key information PDF using the Web Configuration Utility:

1 In your Internet browser, enter your phone's IP address into your browser's address bar.

2 Log in as an **Admin**, enter the default password, and select **Submit**, as shown next.
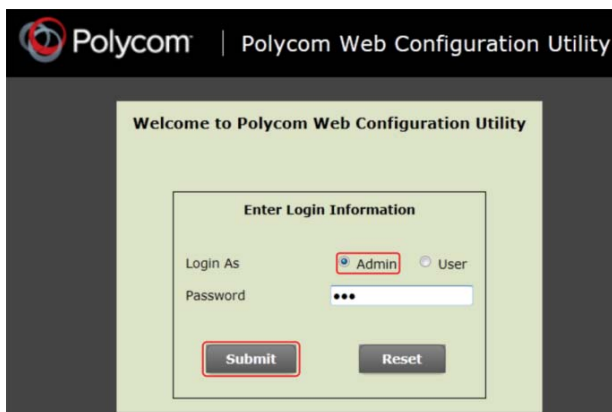
3 Select **Utilities > EM Directory**.

4   Select the expansion module you want to generate a PDF for. For example, EM1 is chosen in the following figure.
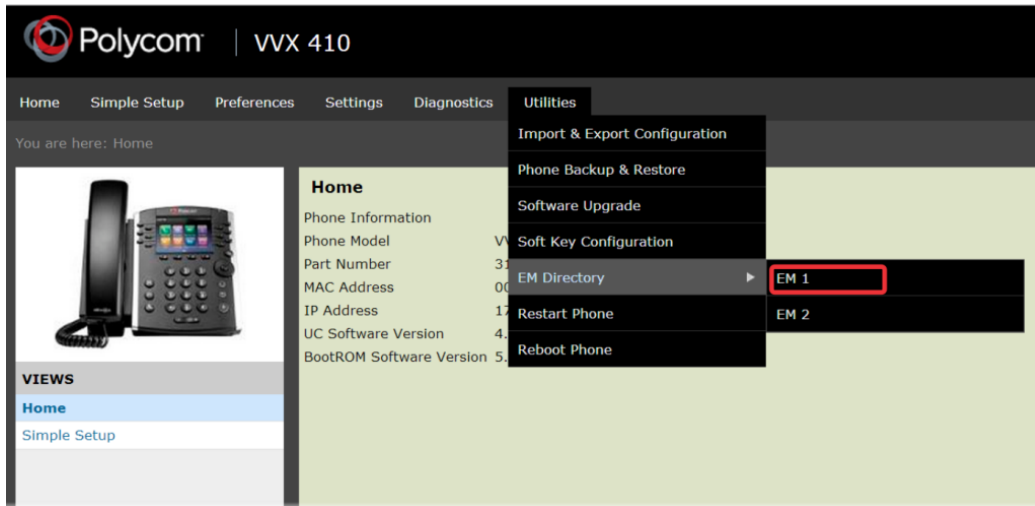
5   In the confirmation dialog, select **Yes** to download the PDF for the configured lines for your expansion module.

6   Select **Save > Open**.

The PDF with the configured line key information for your expansion module displays.

After you download the PDF with configured line key information for your expansion module, you can print the PDF and insert the PDF as the directory card for the expansion module.

Configure Smart Paging

The smart paging feature arranges line key assignments and distributes pages on the VVX Color Expansion Modules based on the number of expansion modules connected to a VVX phone. Smart paging is automatically enabled for color expansion modules connected to VVX phones with UC Software 5.1.0 or later, and is not available on the VVX Expansion Modules with a paper display. Note that when the flexible line key feature is enabled, the expansion module ignores the smart paging configuration and line key assignments display on the designated line key.

> **Note: Line keys are numbered sequentially**
>
> Line keys on VVX phones and expansion modules are numbered sequentially, and the line key numbering on your expansion module depends on how many lines your phone supports. For example, a VVX 600/601 phone supports 16 lines, numbered 1-16. The first line on an expansion module connected to a VVX 600/601 phone is line 17.

The following table lists the configuration parameter you need to enable and disable the smart paging feature.

Configuring Smart Paging

Parameter Function	**template** > parameter
Enable and disable smart paging.	**em.cfg** > up.em.smartpaging.enabled

Example Smart Paging Configuration

In the following example, smart paging is enabled in the `up.em.smartpaging.enabled` parameter in the em.cfg template. By default, the `up.em.smartpaging.enabled` parameter is set to 1, as shown in the following figure.

Example of Smart Paging configuration
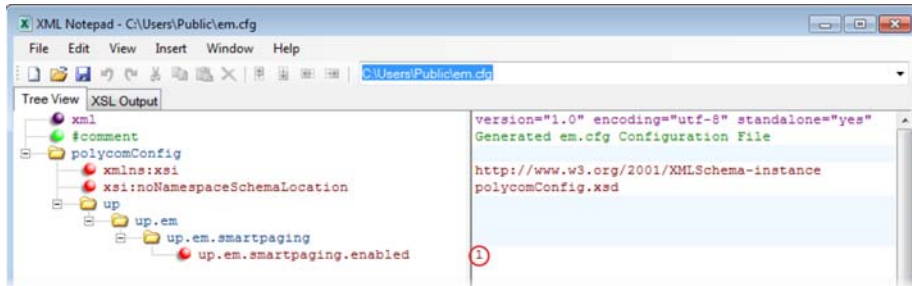
When you enable smart paging, the pages on the color expansion module are distributed across the connected expansion modules, as described in the following scenarios.

> **Note: Smart paging available with multiple expansion modules**
> Smart paging applies only when you connect more than one expansion module to your VVX phone. If you connect one expansion module, the order of pages is sequential even if smart paging is disabled.

● If only one expansion module is connected to the VVX phone, the pages are ordered sequentially on the module: pages 1, 2, and 3.

● If two expansion modules are connected, the pages are ordered non-sequentially across both expansion modules where pages 1, 3, and 4 are on the first expansion module, and pages 2, 5, and 6 are on the second expansion module.

- If you are using three connected expansion modules, the pages are distributed across all modules where pages 1, 4, and 5 are on the first expansion module, pages 2, 6, and 7 are on the second expansion module, and pages 3, 8, and 9 are on the third expansion module.

Pairing and Registering a Polycom VVX D60 Wireless Handset and Base Station

The Polycom® VVX® D60 Wireless Handset provides busy users with the superior voice quality and features of a VVX business media phone that they can use to manage calls when they are away from their desk. You can pair one base station and register up to 5 wireless handsets to the VVX 300 series, 400 series, 500 series, and 600 series phones.

After the base station is paired with the VVX phone, you can register up to five wireless handsets to the base station.

VVX D60 Configuration

The following table lists the configuration parameters you need to configure the VVX D60 feature. Refer to the Configuration Parameters section for more information on these parameters.

Configuring the VVX D60 Accessories

Parameter Function	**template** > parameter
Enable or disable communication and pairing with the VVX D60 Wireless Handset and Base Station accessories.	**features.cfg** > feature.dect.enabled

Configuring the VVX D60 Accessories

Controls the registration index that is used as the default line for outgoing calls placed on the wireless handset without selecting a line first.	**features.cfg** > VVXD60.Handset.X.outGoingLineIndex
Sets the lines that will be accessible from the wireless handset.	**features.cfg** > VVXD60.Handset.X.line.Y
Determines the type of termination that is used for the line.	**reg.cfg** > reg.x.terminationType
Sets the logging detail level for the VVX D60 accessories.	**techsupport.cfg** > log.level.change.dect

Pair the VVX D60 Base Station

You can pair the VVX D60 Base Station and register the Wireless Handset to a VVX phone using the local phone interface or the Web Configuration Utility. You can use the following methods to pair the base station with a VVX business media phone:

● **PC Port Pairing**
● **Automatic Pairing**
● **Manual Pairing**

Base Station initial connection to a VLAN network
After connecting the base station to a LAN port, the base station needs at least one minute to connect to the voice VLAN network and to acquire an IP address. Wait at least one minute after connecting the base station to a LAN port before pairing the base station with a VVX business media phone.

Pair the Base Station via the PC Port using the Web Configuration Utility

When an Ethernet cable is connected to the LAN port on the VVX D60 Base Station and the PC port on a VVX phone, the phone initiates SSH tunneling and pairs with the base station automatically.

To pair the base station via the PC port:

1 Sign into the Web Configuration Utility, and navigate to **Settings > VVX D60 Settings**.
2 Click the **VVX D60 Profile** check box and select **PC Port** in the **Pairing Mode** drop-down.

Pair the Base Station via Automatic Pairing using the Web Configuration Utility

When an Ethernet cable is connected from the Base Station LAN port into a LAN outlet, the phone pairs with the base station automatically.

All base stations on the network are displayed automatically in the VVX Web Configuration Utility as long as the devices are on the same subnetwork and VLAN.

To pair with the base station via Automatic Pairing:

1 Sign into the Web Configuration Utility, and navigate to **Settings > VVX D60 Settings**.
2 Click the **VVX D60 Profile** check box and select **Automatic** in the **Pairing Mode** drop-down.
   A list of discovered base stations and wireless handsets display.
3 Select a base station and click **Pair**.

Pair the Base Station via Manual Pairing using the Web Configuration Utility

When the Ethernet cable is connected from the Base Station LAN port to the VVX PC port or when the Ethernet cable is connected from the Base Station LAN port into a LAN outlet, you can manually enter the Base Station IP address to pair with a VVX phone. Manual pairing enables you to pair the base station with the phone without the base station being on the same subnetwork or VLAN as the VVX phone.

You can use either the Static or DHCP IP address to pair the base station with the phone.

To get the static or DHCP IP address of the base station:

1 Connect an Ethernet cable from the PC port on the base station to an Ethernet port on a computer.
2 On the computer, navigate to **Network and Sharing Center**, then select **Local Area Connection**.
3 Click **Properties**, select **Internet Protocol Version 4 (TCP/IPv4)**, then click **Properties**.
4 Select **Use the following IP address**, then enter the following values into the associated fields:
    ➢ IP Address: **192.168.0.10**
    ➢ Subnet mask: **255.255.255.0**
    ➢ Default Gateway: **192.168.0.1**
5 Click **OK**.
6 In a web browser, enter **https:\\192.168.0.2**.
7 In the Web Configuration Utility, enter the following default credentials:
    ➢ Username: **Polycom**
    ➢ Password: **456**
8 Navigate to **Settings > Network Settings**.
    The IP address of the base station displays in the IP Settings tab.

To pair with the base station via Manual Pairing:

1 Sign into the Web Configuration Utility, and navigate to **Settings > VVX D60 Settings**.
2 Click the **VVX D60 Profile** check box and select **Manual** in the **Pairing Mode** drop-down.
3 Enter the IP address of the base station.
    The base station's information displays.

Pair the Base Station via the PC Port using the Local Phone Interface

When the Ethernet cable is connected from the base station LAN port to the PC port on the VVX phone, the phone pairs with the base station automatically.

To pair with the base station via PC Port Pairing:

1 On the phone, navigate to **Settings > Advanced**, and enter the password.
2 Select **Administration Settings > VVX D60 Configuration**.
3 Select **VVX D60 Profile**, then select **Enable**.
4 On the **VVX D60 Configuration** screen, select **Base Station**, then select **PC Port Pairing**.

Pair the Base Station via Automatic Pairing using the Local Phone Interface

When an Ethernet cable is connected from the Base Station LAN port into a LAN outlet, the phone pairs with the base station automatically.

All base stations on the network are displayed automatically on the VVX phone as long as the devices are on the same subnetwork or VLAN.

To pair with the base station via Automatic Pairing:

1 On the phone, navigate to **Settings > Advanced**, and enter the password.

2 Select **Administration Settings > VVX D60 Configuration**.

3 Select **VVX D60 Profile**, then select **Enable**.

4 On the **VVX D60 Configuration** screen, select **Base Station**, then select **Auto Pairing**.

Pair the Base Station via Manual Pairing using the Local Phone Interface

When the Ethernet cable is connected from the Base Station LAN port to the VVX PC port or when the Ethernet cable is connected from the Base Station LAN port into a LAN outlet, you can manually enter the Base Station IP address to pair with a VVX phone. Manual pairing enables you to pair the base station with the phone without the base station being on the same subnetwork or VLAN as the VVX phone.

You can use either the Static or DHCP IP address to pair the base station with the phone.

To get the static or DHCP IP address of the base station:

1 Connect an Ethernet cable from the PC port on the base station to an Ethernet port on a computer.

2 On the computer, navigate to **Network and Sharing Center**, then select **Local Area Connection**.

3 Click **Properties**, select **Internet Protocol Version 4 (TCP/IPv4)**, then click **Properties**.

4 Select **Use the following IP address**, then enter the following values into the associated fields:

  ➢ IP Address: **192.168.0.10**

  ➢ Subnet mask: **255.255.255.0**

  ➢ Default Gateway: **192.168.0.1**

5 Click **OK**.

6 In a web browser, enter **https:\\192.168.0.2**.

7 In the Web Configuration Utility, enter the following default credentials:

  ➢ Username: **Polycom**

  ➢ Password: **456**

8 Navigate to **Settings > Network Settings**.

  The IP address of the base station displays in the IP Settings tab.

To pair with the base station using manual pairing:

1 On the phone, navigate to **Settings->Advanced**, and enter the password.

2 Select **Administration Settings > VVX D60 Configuration**.

3 Select **VVX D60 Profile**, then select **Enable** and go to the previous menu

4   On the **VVX D60 Configuration** screen, select **Base Station**, then select **Manual Pairing**.

5   Enter the IP address of the base station, then select **Pair**.

    The base station's information displays.

Register a VVX D60 Wireless Handset

After the VVX D60 Base Station is paired with the VVX phone, you can register up to five wireless handsets to the base station. To register the wireless handset with the base station:

To register the wireless handset with the base station:

1   On the wireless handset, navigate to **Settings > Features > Registration**, select **Register**.

2   Press and hold the Find button on the base station for a few seconds.

3   On the wireless handset, confirm the registration with the base station.

Unregister a VVX D60 Wireless Handset

You can unregister a wireless handset from the base station when you need to replace a wireless handset with another one.

To unregister a wireless handset:

1   On the wireless handset, navigate to **Settings > Features > Registration**, select **Deregister**.

2   Confirm you want to unregister the wireless handset.

Setting a Unique Name for the Base Station and Wireless Handset

In the Web Configuration Utility, you can set a unique name for each base station and wireless handset to distinguish between multiple sets of base stations and wireless handsets. You can also set a unique name for the base station and wireless handsets from the local phone interface.

Note that the Intercom feature must be enabled to change the name of a wireless handset. You cannot set a unique name for a wireless handset if the Intercom feature is disabled.

To set a unique name for the base station and registered wireless handsets:

1   In the Web Configuration Utility, navigate to **Settings > VVX D60 Settings**.

2   Under **Base Station Settings**, enter a unique name in the **Name** field.

3   Under **Handset Settings**, enter a unique name in the **Display Name** field for each registered handset.

Assign Lines to the VVX D60 Wireless Handset

After you have paired the base station to a VVX phone and registered wireless handsets to the base station, you can assign lines to each wireless handset. You can assign up to five lines to each wireless handset.

When assigning lines, keep the following in mind:

● The first line is assigned to the VVX phone.

● For Private Lines, you can assign each line to the VVX phone or the Wireless Handset or both.

- For Shared lines (SCA/SLA), you can assign each line only to one device: VVX phone or Wireless Handset.

Assign Lines using the Web Configuration Utility

You can assign lines to the wireless handset using the Web Configuration Utility.

To assign lines to a VVX wireless handset:

1 In the Web Configuration Utility, navigate to **Settings > VVX D60 Settings**.
2 In the Handset Settings section, click **Map Lines**.
3 Choose the lines you want to map to a registered wireless handset.
4 Click **Update**.

Assign Lines using the Phone Interface

You can assign lines to the wireless handset from the Advanced settings menu on the VVX phone.

To assign lines to a VVX wireless handset:

1 On the phone, navigate to **Settings > Advanced**, then enter your password.
2 Select **Administration Settings > VVX D60 Configuration > Map Lines**.
3 Choose a line, then choose a registered wireless handset for the line.

Update the VVX D60 Wireless Handset Software

When you update the VVX host phone with the latest supported software version using the master configuration file that includes the file path to the dect.ld, the software on the base station and wireless handsets update automatically within two minutes after they are paired and registered with the VVX business media phone. The base station updates first, then each wireless handset is updated sequentially with the first registered handset updating first, followed by each remaining handset.

Refer to the section Configure with the Master Configuration File for more information on the fields in the master configuration file.

To update the wireless handset software:

1 Place the handset in the base station or charging cradle, and ensure the handset battery is charged to at least 50%.
2 When prompted, accept the update notification.

   If you do not accept the update notification, the wireless handset will begin the update 20 seconds after the notification displays.

If the base station or any of the wireless handsets fail to update, refer to the section Wireless Handset Software Upgrade Issues for troubleshooting tips.

Update the Wireless Handset Software Manually

If the software update notification does not display on the wireless handset within five minutes of registering the wireless handset, you can check for configuration updates and manually update the software from the VVX host phone.

To update the wireless handset software manually:

1   Place the handset in the base station or charging cradle, and ensure the handset battery is charged to at least 50%.

2   On the VVX host phone, navigate to Settings > Basic > Update Configuration.

    If there is a software update available, the wireless handsets update sequentially with the first registered handset updating first.

Configuring VVX D60 Network Settings

By default, you cannot edit network settings for the VVX D60 Base Station. To make any changes to network settings for the base station, you need to place the base station into recovery mode. After the base station is in recovery mode, you can use the Web Configuration Utility to make changes to the base station's network settings.

Place the VVX D60 Base Station into Recovery Mode

Place the base station into recovery mode to make any changes to network settings for the base station.

To place the base station into recovery mode:

»   Using a paper clip, press and hold the Reset button on the back of the base station for eight seconds.

    The base station restarts in recovery mode.

Configure VVX D60 Network Settings

While the VVX D60 base station is in recovery mode, you can use the Web Configuration Utility to update network settings for the base station.

To configure VVX D60 network settings:

1   In a web browser, enter **https:\\192.168.0.2**.

2   In the Web Configuration Utility, enter the following default credentials:

    ➢   Username: **Polycom**

    ➢   Password: **456**

3   Navigate to **Settings > Network Settings**.

Enable the Power-Saving Feature

The VVX 500, 600, and 1500 phones support a power-saving feature that is enabled by default. This feature has a number of options you can configure:

●   Turn on the phone's power-saving feature during non-working hours and working hours. When you enable power-saving mode on VVX 500 and 600, the phone display screen does not automatically turn back on after going idle.

●   If you want to turn on power-saving during non-working hours, you can configure the power-saving feature around your work schedule.

●   On the VVX 1500 only, use the `powerSaving.userDetectionSensitivity.*` parameters to configure the sensitivity of the built-in motion detection system and an idle time after which the phone enters the power-saving mode.

When you enable power-saving mode and the phone is in low power state, the red LED indicator flashes at three second intervals to show that the phone still has power.

Power-Saving Parameters

Parameter Function	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.*
Set the phone's motion detection sensitivity.	**site.cfg** > powerSaving.userDetectionSensitivity.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Example Power-Saving Configuration

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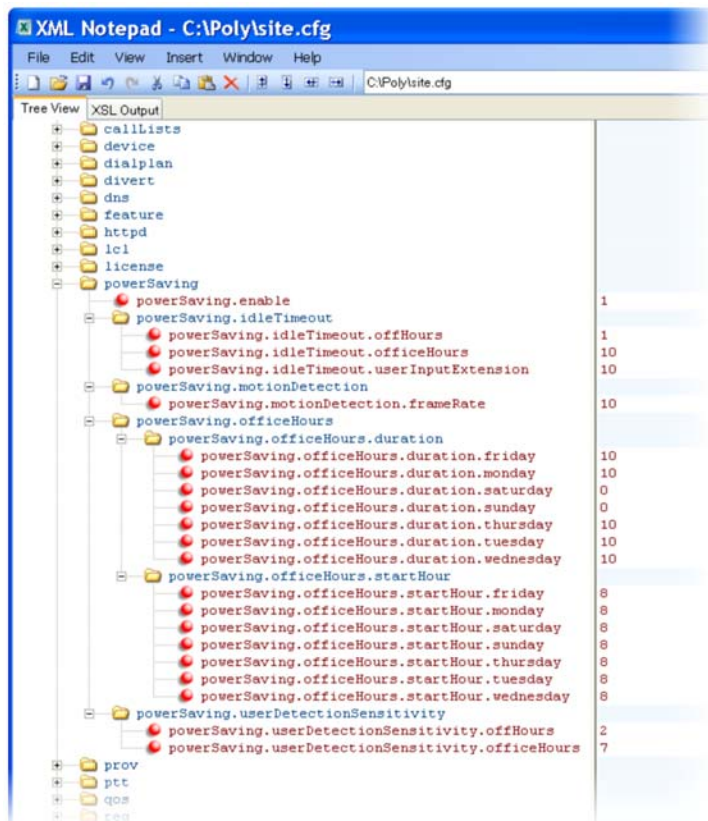
Enable Polycom Desktop Connector Integration

With the Polycom® Desktop Connector™ application installed on a computer, you can use your mouse and keyboard to enter information and navigate screens on your VVX phone running Polycom UC Software 4.0.1 or later. This feature enables users to enter phone numbers or to select screen objects without having to use the phone's keypad or touch screen. To use this feature, the phone and computer must be on the same network or directly connected through the phone's PC port.

You need to download and install the Polycom Desktop Connector application. The Polycom Desktop Connector is compatible with computers running Microsoft® Windows XP®, Windows Vista®, and Windows® 7.

Once Polycom Desktop Connector is installed, you need to pair the VVX phone and the computer (to configure the connection). If they are directly connected, there is no need to enter the VVX phone's IP address; just press the Reconnect soft key. If they are connected through a switch or hub, enter the computer's IP address using the phone's user interface, and press the Reconnect soft key. You can also change the configuration by editing the phone's configuration parameters shown in the following table or by using the Web Configuration Utility.

> **Web Info: Installing and enabling the Polycom Desktop Connector application**
> For details on how to install Polycom Desktop Connector application and enable it for use on VVX phones, see the latest *Polycom VVX Business Media Phones User Guide* at Latest Polycom UC Software Release.

Enable Polycom Desktop Connector Integration

Parameter Function	template > parameter
Turn the Desktop Connector on or off for administrators.	**applications.cfg** > apps.ucdesktop.adminEnabled
Specify the user name of the user's computer.	**applications.cfg** > apps.ucdesktop.desktopUserName
Turn the Desktop Connector on or off for users.	**applications.cfg** > apps.ucdesktop.enabled
Specify if the phone is positioned to the left or right of your computer.	**applications.cfg** > apps.ucdesktop.orientation
Specify the server address of the user's computer.	**applications.cfg** > apps.ucdesktop.ServerAddress
Specify the server port number for the connection.	**applications.cfg** > apps.ucdesktop.ServerPort

Example Desktop Connector Configuration

To use the Polycom Desktop Connector, ensure that the `apps.ucdesktop.adminEnabled` in the applications.cfg template parameter is enabled, as shown next. By default, the parameter is enabled.
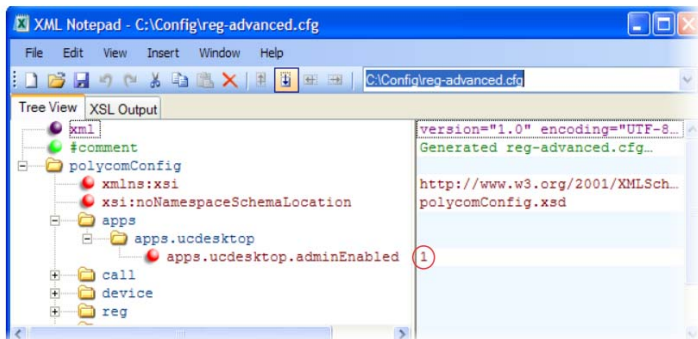
The following illustration shows the parameters in applications.cfg that you need to configure to use the Polycom Desktop Connector on your phones. You'll have to enable the feature, as well as specify a user name, server address and port, and specify the phone's position relative to your computer.
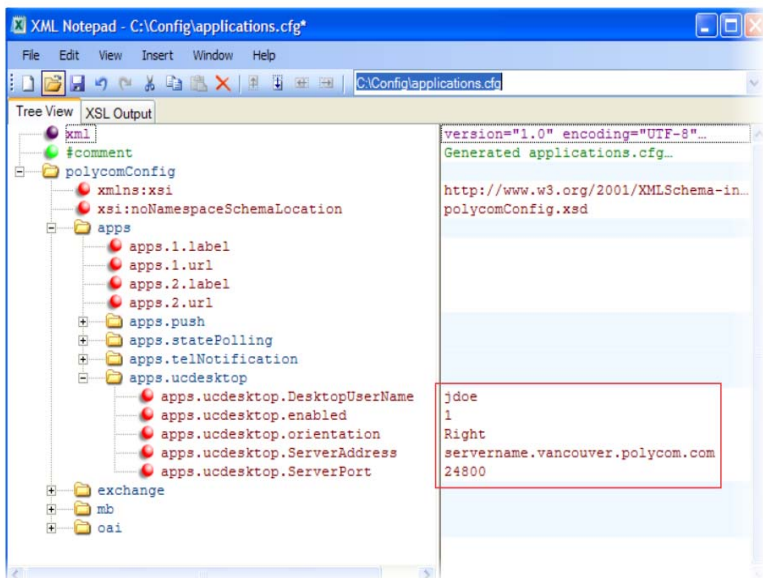
Customize Phone Audio

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. This section provides information on configuring phone audio.

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 Polycom phones. Use these features and options to optimize the conditions of your organization's phone network system.

Polycom Acoustic Fence™

Available on all VVX business media phones, the Polycom Acoustic Fence feature enables you to enhance background noise suppression when using the phone handset or a headset. This feature is particularly useful in call center environments where background noise can impact far-end audio quality.

Configure Noise Suppression Parameters

Parameter Function	template > parameter
Enable or disable noise suppression for headsets.	**features.cfg** > voice.ns.hd.enable
Enable or disable Acoustic Fence noise suppression for headsets.	**features.cfg** > voice.ns.hd.enhanced
Increase or decrease the noise suppression threshold on headsets.	**features.cfg** > voice.ns.hd.nonStationaryThresh
Enable or disable noise suppression for handsets.	**features.cfg** > voice.ns.hs.enable
Enable or disable Acoustic Fence noise suppression for handsets.	**features.cfg** > voice.ns.hs.enhanced
Increase or decrease the noise suppression threshold for handsets.	**features.cfg** > voice.ns.hs.nonStationaryThresh

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)

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

Customize Audio Sound Effects

Parameter Function	**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.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

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).
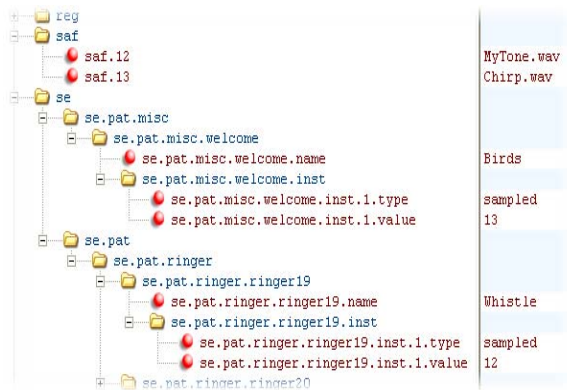
The following illustration shows the custom ring tone Whistle as it displays on the phone menu:

Adjust Context Sensitive Volume Control

The parameters shown in the following table enable you to adjust the volume of phone sound effects—such as the ringer and the volume of receiving call audio—separately for the speakerphone, handset, and headset. While transmit levels are fixed according to the TIA/EIA-810-A standard, you can adjust the receive volume. The receiving volume of the handset and headset resets after each call to comply with regulatory requirements. The hands free speakerphone volume level remains at the same level as the previous call.

Context Sensitive Volume Control

Parameter Function	template > parameter
Specify if a Bluetooth headset should be used for every call (VVX 600/601 only).	**site.cfg** > voice.volume.persist.bluetooth.headset
Specify if the volume level of the handset, headset, and speakerphone should reset after each call.	**site.cfg** > voice.volume.persist.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Use Voice Activity Detection

The purpose of voice activity detection (VAD) 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 following table 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 Parameters

Parameter Function	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

Control Comfort Noise Payload Packets

When enabled, the Comfort Noise payload type is negotiated in SDP with the default of 13 for 8 KHz codecs and a configurable value between 96 and 127 for 16 KHz codecs.

The following table lists the parameters you can use to enable Comfort Noise Control.

Comfort Noise Parameters

Parameter Function	**template** > parameter
Specify if the support for Comfort Noise in the SDP body of the INVITE message is published with the supported comfort noise payloads included in the media line for audio.	**site.cfg** > voice.CNControl
Specify the dynamic payload type used for Comfort Noise RTP packets.	**site.cfg** > voice.CN16KPayload

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 following table 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

Parameter Function	**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

Set 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 following table to set up this feature.

DTMF Event RTP Payload

Parameter Function	**template** > parameter
Specify if the phone 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 Polycom phone uses advanced acoustic echo cancellation (AEC) for handsfree operation using the speakerphone. See the table Audio Codec Priority for a list of audio codecs available for each phone and their priority. The phone also supports headset echo cancellation. The phones use 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 Codec Support

The following table details the audio codec support and priority for Polycom phones.

Note the following limitations when using the Opus codec:

- VVX 301, 311, 401, 411, 500, and 600 business media phones support a single Opus stream. You can establish only one call at a time using Opus codec on these phones.
- VVX 500 and 600 do not support video when using Opus.
- VVX 500 and 600 do not support local conferences when using Opus.
- Opus is not compatible with G.729 and iLBC. If you set Opus to highest priority, G.729 and iLBC are not published; if you set G.729 and iLBC to highest priority, Opus is not published.

> **Settings: Video disables G.722.1C codec**
> On the VVX 500/501 and 600/601, when you enable video, the G.722.1C codec is disabled.

Audio Codec Priority

Phone	Supported Audio Codecs	Priority
VVX 101, 201	G.711μ-law	6
	G.711a-law	7
	G.722	4
	G.722.1 (32kbps)	5
	G.729AB	8
	iLBC (13.33kbps, 15.2kbps)	0, 0

Audio Codec Priority (continued)

VVX 300/301, 310/311, 400/401, 410/411 *Note: VVX 301, 311, 401, 411 support a single Opus stream. VVX 300, 310, 400, 410 do not support Opus.	G.711µ-law	6
	G.711a-law	7
	G.722	4
	G.722.1 (32kbps)	5
	G.729AB	8
	iLBC (13.33kbps, 15.2kbps)	0, 0
	Opus*	0
VVX 500/501, 600/601 *Note: VVX 500 and 600 support a single Opus stream. VVX 501 and 601 do not support Opus.	G.711 µ -law	6
	G.711a-law	7
	G.722	4
	G.722.1 (32kbps)	5
	G.722.1C (48kbps)	2
	G.729AB	8
	Opus*	0
	iLBC (13.33kbps, 15.2kbps)	0, 0
VVX 1500	G.711 µ -law	6
	G.711a-law	7
	G.719 (64kbps)	0
	G.722	4
	G.722.1 (32kbps)	5
	G.722.1C (48kbps)	2
	G.729AB	8
	Siren14 (48kbps)	3
	iLBC (13.33kbps, 15.2kbps)	0, 0

Audio Codec Priority (continued)

SoundStructure VoIP Interface	G.711 μ -law	6
	G.711a-law	7
	G.722	4
	G.722.1 (32kbps)	5
	G.722.1C (48kbps)	2
	G.729AB	8
	iLBC (13.33kbps, 15.2kbps)	0, 0

The table following summarizes the audio codecs supported on Polycom phones.

Audio Codec Specifications

Algorithm	Reference	Raw Bit Rate	IP Bit Rate	Sample Rate	Default Payload Size	Effective Audio Bandwidth
G.711 μ -law	RFC 1890	64 Kbps	80 Kbps	8 Ksps	20 ms	3.5 KHz
G.711 a-law	RFC 1890	64 Kbps	80 Kbps	8 Ksps	20 ms	3.5 KHz
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 3551	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
Opus	RFC 6716	8 - 24 Kbps	24 - 40 Kbps	8 Ksps 16 Ksps	20 ms	3.5 KHz 7 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

Audio Codec Specifications

Algorithm	Reference	Raw Bit Rate	IP Bit Rate	Sample Rate	Default Payload Size	Effective Audio Bandwidth
Siren14	SIREN14	24 Kbps 32 Kbps 48 Kbps	40 Kbps 48 Kbps 64 Kbps	32 Ksps	20 ms	14 KHz
Siren22	SIREN22	32 Kbps 48 Kbps 64 Kbps	48 Kbps 64 Kbps 80 Kbps	48 Ksps	20 ms	22 KHz
iLBC	RFC 3951	13.33 Kbps 15.2 Kbps	31.2 Kbps 24 Kbps	8 Ksps	30 ms 20 ms	3.5 KHz

[1] Per RFC 3551. Even though the actual sampling rate for G.722 audio is 16,000 Hz (16 ksps), the RTP clock rate advertised for the G.722 payload format is 8,000 Hz because that value was erroneously assigned in RFC 1890 and must remain unchanged for backward compatibility.

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 following table to specify the priority for audio codecs on your Polycom phones.

Audio Codec Priorities

Parameter Function	**template** > parameter
To specify the priority for a codec.	**site.cfg** > voice.codecPref.<nameOfCodec>

Set 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 following table 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)

Parameter Function	**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.*

IP Type-of-Service (ToS)

Set the IP header bits for RTP video.   **site.cfg** > qos.ip.rtp.video.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Set IEEE 802.1p/Q

The phone tags 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 Modify DHCP Settings)

Use the following table 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

Parameter Function	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

Configure 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 Session initiation Protocol Package for Voice Quality Reporting Event. The metrics can also be sent as SIP PUBLISH messages to a central voice quality report collector.

You can use Real Time Control Protocol Extended Report (RTCP XR) to report voice quality metrics to remote endpoints. This feature is available on RealPresence Trio solution and VVX phones and supports RFC6035 compliance as well as draft implementation for voice quality reporting.

Voice quality monitoring metrics are supported on the VVX phones and the RealPresence Trio 8800.

You require a license key to activate the VQMon feature on the following VVX business media phones:

- 101, 201, 300, 301, 310, 311, 400, 401, 410, 411

The following phones do not require a license key to activate the VQMon feature:

- 500, 600, 1500, Polycom RealPresence Trio

This feature is available for open SIP and is not available with Lync Server or Skype for Business. 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 the following table. 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)

Parameter Function	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
Specify the standards compliance.	**features.cfg** > voice.qualityMonitoring.rfc6035.enable
Enable or disables re-registration on failover.	**features.cfg** > voice.qualityMonitoring.failover.enable
Specify the device location with a valid location string.	**features.cfg** > voice.qualityMonitoring.location

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Configure Audible Ringer Location

You can choose where all audio alerts, including incoming call alerts, are played out on devices running UC Software 3.3.0 or later. Use the following table to specify where you hear audio. You can specify the audio to play from the handsfree speakerphone (default), the handset, the headset, or the active location. If you choose the active location, audio alerts play out through the handset or headset if they are off hook. Otherwise, alerts play through the speakerphone.

Audible Ringer Location

Parameter Function	template > parameter
Specify where audio alerts play out from.	**reg-advanced.cfg** > se.destination

Use Bluetooth Headset Support

You can use Bluetooth v2.1 headsets with VVX 600/601 business media phones. To use a Bluetooth headset, you need to enable the Bluetooth headset feature and turn on the Bluetooth radio, as shown in the following table.

Troubleshooting: Using a Bluetooth headset affects my phone's voice quality
You may not experience the highest voice quality if you use a Bluetooth headset while the 2.4 GHz band is enabled or while you are in an environment with many other Bluetooth devices. This possible loss in voice quality is due to inherent limitations with Bluetooth technology.

Bluetooth Headset Support

Parameter Function	template > parameter
To enable or disable the Bluetooth headset feature.	**features.cfg** > feature.bluetooth.enabled
To turn the Bluetooth radio (transmitter/receiver) on or off.	**features.cfg** > bluetooth.radioOn

Set Up Phone Video

After you set up Polycom phones on your network with the default configuration, users can place and answer video calls. Polycom's Open SIP UC Software enables you to make custom configurations to optimize video calling. This section provides information on configuring phone video.

The Polycom VVX 500/501, 600/601 and 1500 phones using the Polycom VVX Camera support transmission and reception of high quality video images. Polycom Open SIP video is compatible with RFC 3984 - RTP Payload Format for H.264 video, RFC 4629 - RTP Payload Format for ITU-T Rec. H.263 Video, and RFC 5168 - XML Schema for Media Control.

Configure Video Transmission

By default, at the start of a video call, the VVX 1500 and VVX phones using the VVX Camera transmit an RTP encapsulated video stream with images captured from the local camera. Users can stop and start video transmission by pressing the Video key, and then selecting the Stop or Start soft key.

You can configure:

- Video Transmission Parameters
- Video and Camera View Parameters
- Video Camera Parameters

Use the parameters in the following table to configure video transmission.

Video Transmission Parameters

Parameter Function	template > parameter
Specify if video calls should use a full screen layout.	**video.cfg** > video.autoFullScreen
Specify when video transmission should start in a call.	**video.cfg** > video.autoStartVideoTx
Set the call rate for a video call (can be changed on the phone).	**video.cfg** > video.callRate
Specify whether the phone is forced to send RTCP feedback messages to request fast update I-frames for video calls.	**video.cfg** > video.forceRtcpVideoCodecControl
Set the maximum call rate for a video call (the maximum rate set from the phone cannot exceed this).	**video.cfg** > video.maxCallRate
Specify the quality of video to be shown in a call or conference.	**video.cfg** > video.quality

You can use the parameters in the following table to set the video and local camera view settings.

Video and Camera View Parameters

Parameter Function	template > parameter
Specify the view of the video window in normal viewing mode.	**video.cfg** > video.screenMode
Specify the view of the video window in full screen viewing mode.	**video.cfg** > video.screenModeFS

Video and Camera View Parameters (continued)

Specify if the local camera view is shown in the full screen layout.	**video.cfg** > video.localCameraView.fullscreen.enabled
Determine how the local camera view is shown.	**video.cfg** > video.localCameraView.fullscreen.mode

You can use the parameters in the following table to configure the video camera.

Video Camera Parameters

Parameter Function	**template** > parameter
Set the brightness level.	**video.cfg** > video.camera.brightness
Set the contrast level.	**video.cfg** > video.camera.contrast
Specify if flicker avoidance is automatic, suited for Europe/Asia, or North America.	**video.cfg** > video.camera.flickerAvoidance
Set the frame rate.	**video.cfg** > video.camera.frameRate
Set the saturation level.	**video.cfg** > video.camera.saturation
Set the sharpness level.	**video.cfg** > video.camera.sharpness

Supported Video Codecs

See the following table for a summary of the VVX phone's video codec support.

Video Codec Specifications

Algorithm	MIME Type	Frame Size	Bit Rate (kbps)	Frame Rate (fps)
H.261	H261/90000	Tx Frame size: CIF, QCIF, SQCIF RX Frame size: CIF, QCIF	64 to 768	5 to 30
H.263	H263/90000, H263-1998/90000	Tx Frame size:CIF, QCIF Rx Frame size:CIF, QCIF, SQCIF, QVGA, SVGA, SIF	64 to 768 kbps	5 to 30
H.264	H264/90000	Tx Frame size:CIF, QCIF Rx Frame size:CIF, QCIF, SQCIF, QVGA, SVGA, SIF	64 to 768	5 to 30

You can configure the parameters in the following table to prioritize and adjust the video codecs that your VVX phones use.

Video Codec Parameters

Parameter Function	**template** > parameter
Prioritize the video codecs from 1 to 4.	**video.cfg** > video.codecPref.*
Adjust the parameters for the H261, H263, H2631998, and H264 codec profiles.	**video.cfg** > video.profile.<codec>.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Configure H.323 Protocol

As of SIP 3.2.2, the VVX 1500 phone and VVX camera-enabled VVX 500/501 and 600/601 phones support telephony signaling via the H.323 protocols. This protocol enables direct communication with H.323 endpoints, gatekeepers, call servers, media servers, and signaling gateways.

> **Note: Activating H.323 video**
> You need a license key to activate H.323 video on your VVX 1500 phone; the license is installed on the VVX 1500D. For more information, contact your Certified Polycom Channel Partner.

The VVX 500/501, 600/601, and 1500 can support SIP and H.323 signaling simultaneously, and you can bridge both types of calls during multi-party conference calls. The phone can automatically detect the correct or optimal signaling protocol when dialing a call from the contact directory or the corporate directory. While SIP supports server redundancy and several transport options, only a single configured H.323 gatekeeper address per phone is supported. The phone does not require H.323 gatekeepers, but use then if available. If an H.323 gatekeeper is not configured or is unavailable, you can still enable your phone to make H.323 calls.

Support of the SIP protocol for telephony signaling can be disabled on the VVX 500/501, 600/601, and 1500 such that all calls route via the H.323 protocol.

This section provides detailed information on:

- Supported H.323 Video Standards
- Supported Video Interoperability
- Use the H.323 Protocol
- FQDN Support for H.323 Gatekeeper Failover

For a list of all H.323 parameters, see the following table.

H.323 Protocol Parameters

Parameter Function	**template** > parameter
Specify if the user is presented with protocol routing choices.	**reg-advanced.cfg** and **site.cfg**> up.manualProtocolRouting
Set soft keys for protocol routing.	**reg-advanced.cfg** and **site.cfg** > up.manualProtocolRouting.softKeys

H.323 Protocol Parameters (continued)

Enable or disable auto-answer for all H.323 calls.	**reg-advanced.cfg** and **h323.cfg** > call.autoAnswer.H323
Specify if the phone can make calls using H.323 even if an H.323 gatekeeper is not configured or is unavailable.	**sip-interop.cfg** > call.enableOnNotRegistered
Specify if video should begin immediately after a call is auto-answered.	**reg-advanced.cfg** > call.autoAnswer.videoMute
Specify whether SIP or H.323 is the preferred call protocol.	**video.cfg** > call.autoRouting.preferredProtocol
Specify if calls should be routed by line or by protocol.	**sip-interop.cfg** > call.autoRouting.preference
Enable or disable H.323 signaling for the line registration.	**sip-interop.cfg** > reg.x.protocol.H323
Specify the H.323 server settings for a specific registration.	**site.cfg** > reg.x.server.H323.*
Specify the H.323 protocol settings.	**h323.cfg** > voIpProt.H323.*
Specify the H.323 server settings.	**h323.cfg** > voIpProt.server.H323.*
Configure the H.323 media encryption parameters.	**site.cfg** > sec.H235.mediaEncryption.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Supported H.323 Video Standards

The following table lists the standards the H.323 feature supports.

Supported Video Standards

Standard	Description
ITU-T Recommendation H.323 (2003)	Packet-based multimedia communications systems
ITU-T Recommendation Q.931 (1998)	ISDN user-network interface layer 3 specification for basic call control
ITU-T Recommendation H.225.0 (2003)	Call signaling protocols and media stream packetization for packet-based multimedia communications systems
ITU-T Recommendation H.245 (5/2003)	Control protocol for multimedia communication
ITU-T Recommendation H.235.0 - H.235.9 (2005)	Security and encryption for H Series (H.323 and other H.245 based) multimedia terminals

Supported Video Interoperability

Polycom endpoints/bridges/call servers (or gatekeepers)/media servers support the video calls listed in the following table.

Supported Polycom Product Interoperability

Polycom Product	Protocol	Software Version
Polycom HDX® 9000 series	SIP/ISDN/H.323	SW 2.6.0
Polycom HDX 8000 series	SIP/ISDN/H.323	SW 2.6.0
Polycom HDX7000 series	SIP/ISDN/H.323	SW 2.6.0
Polycom HDX 6000	SIP/ISDN/H.323	SW 2.6.0
Polycom HDX 4000 series	SIP/ISDN/H.323	SW 2.6.0
Polycom RMX® 2000	H.323	SW 4.0.2.7
Polycom Quality Definition Experience™ (QDX™)	H.323	SW 4.0, 4.0.1
Polycom RMX 1000	H.323	SW 1.1.1.8787
Polycom RMX 2000	H.323	SW 5.0.1.24, 6.0
Polycom RSS™	H.323	SW 6.0
Polycom VBP™ 6400-ST series	H.323	SW 9.1.5.1
Polycom VBP 5300-ST series	H.323	SW 9.1.5.1
Polycom VBP 5300-E series	H.323	SW 9.1.5.1
Polycom VBP 4350 series	H.323	SW 9.1.5.1
Polycom VBP 200	H.323	SW 9.5.2
Polycom VSX® 8000	SIP/ISDN/H.323	SW 9.0.6
Polycom VSX 7000s and VSX 7000e	SIP/ISDN/H.323	SW 9.0.6
Polycom VSX 6000 and 6000a	SIP/ISDN/H.323	SW 9.0.5.1
Polycom VSX 5000	SIP/ISDN/H.323	SW 9.0.5.1
Polycom VSX 3000	SIP/ISDN/H.323	SW 9.0.5.1
Polycom V700™	SIP/ISDN/H.323	SW 9.0.5.1
Polycom V500™	SIP/ISDN/H.323	SW 9.0.5.1
RealPresence® Group Series 300	H.239/H.261/H.263\H.264/SIP/TIP	SW 4.1.1
RealPresence Group Series 500	H.239/H.261/H.263\H.264/SIP/TIP	SW 4.1.1
RealPresence Group Series 700	H.239/H.261/H.263\H.264/SIP/TIP	SW 4.1.1

Web Info: Viewing an updated list of Polycom video support with third-party products

See the *UC Software Release Notes* on the Latest Polycom UC Software Release page for the latest list of supported Polycom endpoints/bridges/call servers (or gatekeepers)/media servers and any supported third party products. Any issues (and possible workarounds) with any of the above-mentioned products are also documented in the Release Notes.

Use the H.323 Protocol

The following information should be noted:

- If the phone has only the H.323 protocol enabled, it cannot be used to answer SIP calls.
- If the phone has only the SIP protocol enabled, it cannot be used to answer H.323 calls.
- If both SIP and H.323 protocols are disabled by mistake, the phone continues to work as a SIP-only phone; however, the phone is not registered (you are able to send and receive SIP URL calls).
- The phone stores the protocol used to place a call in the placed call list.
- The protocol to be used when placing a call from the user's local contact directory is unspecified by default. The user can select SIP or H.323.
- The protocol that is used when placing a call from the user's corporate directory depends on the order of the attributes in the corporate directory. If only `SIP_address` is defined, then the SIP protocol is used. If only `H323_address` is defined, then the H.323 protocol is used. If both are defined, then the one that is defined first is used. For example, if `dir.corp.attribute.4.type` is `SIP_address` and `dir.corp.attribute.5.type` is `H323_address`, then the SIP protocol is used.
- By default, when more than one protocol is available, each protocol displays as a soft key and the user can choose which protocol to use.
- Calls made using H.323 cannot be forwarded or transferred.
  - ➢ The Transfer and Forward soft keys do not display during an H.323 call on a VVX 500/501, 600/601, or 1500 phone. The Forward soft key does not display on the idle screen on a VVX 500/501, 600/601, or 1500 phone if the primary line is an H.323 line.
  - ➢ If a VVX 500/501, 600/601, or 1500 user presses the Transfer soft key during an H.323 call, no action is taken.
  - ➢ The auto-divert field in the local contact directory entry is ignored when a call is placed to that contact using H.323.
  - ➢ If a conference host ends a three-way conference call and one of the parties is connected by H.323, that party is not transferred to the other party that was part of the conference call.

Example H.323 Configuration

The following illustrates an example of a sip-h323.cfg file and the parameters you need to configure.

```
phone
  voIpProt
    SIP
        voIpProt.SIP.enable              1
    H323
        voIpProt.H323.enable             1
  dialplan
    digitmap
        dialplan.digitmap                0xxxS|33xxH
  user_preferences
      up.manualProtocolRouting           1
      up.manualProtocolRouting.softKeys  1
  call
      call.autoAnswer.SIP                0
      call.autoAnswer.H323               1
      call.autoAnswer.micMute            1
      call.autoAnswer.videoMute          0
      call.autoRouting.preference        line
      call.autoRouting.preferredProtocol SIP
      call.autoOffHook.3.protocol        SIP
  reg
      reg.1.address                      1301
      reg.1.server.1.address             sipserver.polycom.com
      reg.1.protocol.SIP                 1
      reg.1.protocol.H323                0
      reg.1.label                        1301S
      reg.2.address                      1302
      reg.2.server.1.address             172.88.2.123
      reg.2.protocol.SIP                 0
      reg.2.protocol.H323                1
      reg.2.label                        1302H
      reg.3.address                      1303
      reg.3.server.1.address             sipserver.polycom.com
      reg.3.server.2.address             172.88.2.123
      reg.3.protocol.SIP                 1
      reg.3.protocol.H323                1
      reg.3.label                        1303D
```

Use these parameters:

- To configure SIP and H.323 protocols
- To set up a SIP and H.323 dial plan

  Numbers with the format 0xxx are placed on a SIP line and numbers with the format 33xx are placed on an H.323 line.

- To set up manual protocol routing using soft keys

  If the protocol to use to place a call cannot be determined, the **Use SIP** and **Use H.323** soft keys display, and you must select one for the call to be placed.

- To configure auto-answering on H.323 calls only.
- To set the preferred protocol to SIP.
- To set to configure one SIP line, one H.323 line, and a dual protocol line—both SIP and H.323 can be used.
- To set the preferred protocol for off-hook calls on the third (dual protocol) line to SIP.

FQDN Support for H.323 Gatekeeper Failover

This enhancement, available only for registration failover scenarios, enables fully qualified domain name (FQDN) configuration for H.323 Gatekeeper. Gatekeeper IP addresses are resolved from a DNS server when the Gatekeeper sends a DNS A query or through the local static cache. This enhancement supports a maximum of two IP addresses based on the DNS response irrespective of the number of records received. Note that this enhancement does not apply if you are using the parameter `voIpProt.H323.autoGateKeeperDiscovery` for autodiscovery.

Toggle Between Audio-only or Audio-Video Calls

When this feature is enabled on the VVX 1500, and VVX camera-enabled VVX 500/501 and 600/601 business media phones, a soft key displays to toggle calls between audio-only or audio-video.

Note that this feature applies only to outbound calls from your phone; incoming video calls to your phone are answered using video even when you set the feature to use audio-only.

When you enable this feature using `feature.audioVideoToggle.enabled`, calls are audio-only by default, and you must toggle the call to use audio-video before the call begins. After a video call has ended, the phone returns to audio-only.

If you want a call mode setting to persist until users manually change the call mode, also enable `audioVideoToggle.callMode.persistent`. Use the following table to locate available parameters.

Audio and Video Toggle Parameters

Parameter Function	template > parameter
Enable or disable the audio/video toggle feature.	**features.cfg** > feature.audioVideoToggle.enabled
Enable or disable the last call mode set by the user.	**video.cfg** > audioVideoToggle.callMode.persistent
Allow the user to select the call mode to use when using SIP protocol only.	**video.cfg** > video.callMode.default
Begin video to the far side when you start a call.	video.autoStartVideoTx
Select the outbound call mode. To use this parameter, you must enable `feature.audioVideoToggle.enabled.`	video.callMode.default

Configure Phone Security

After you set up Polycom devices on your network with the default configuration, users can place and answer calls. Polycom UC Software enables you to make custom configurations to optimize security settings. This section provides information on configuring phone security.

Lock the Web Configuration Utility after Failed Login Attempts

For additional security, you can lock access to the Web Configuration Utility after failed user login attempts and configure a period of time after which a user can attempt to log in again.

Lock Web Configuration Utility Parameters

Parameter template	Permitted Values
`httpd.cfg.lockWebUI.enable` site.cfg	1 (default) - Enable the Web Configuration Login Lock feature. 0 - Disable the Web Configuration Login Lock feature.
`httpd.cfg.lockWebUI.lockOutDuration` site.cfg	60 seconds (default) - The period of time the user is locked out of the Web Configuration Utility. The user can try logging in again after this time. 60 - 300 seconds
`httpd.cfg.lockWebUI.noOfInvalidAttempts` site.cfg	5 (default) - After five failed login attempts, the user is locked out of the Web Configuration Utility. 3 - 20 - Specify the maximum number of failed login attempts after which the user is locked out of the Web Configuration Utility.

`httpd.cfg.lockWebUI.noOfInvalidAttempts`
`Duration`
site.cfg
	60 seconds (default) - After a user reaches the maximum failed login attempts within 60 seconds, the user is locked out of the Web Configuration Utility.
	60 - 300 seconds - After a user reaches the maximum failed login attempts within this time duration, the user is locked out of the Web Configuration Utility. The user can try logging in again after the lock-out duration set by `httpd.cfg.lockWebUI.lockOutDuration`.

SIP Instance Support

In environments where multiple phones are registered using the same address of record (AOR), the phones are identified by their IP address. However, firewalls set up in these environments can change the IP addresses regularly for security purposes. This feature provides support for SIP instance to identify individual phones instead of using IP addresses. This feature complies with RFC 3840.

Support for SIP instance is available on VVX 300 series, 400 series, 500 series, 600 series, and 1500 business media phones. The parameter reg.x.gruu provide a unique identification (a contact address) to a specific user agent (UA) instance. This helps to route the request to the UA instance and is required in cases in which the REFER request must be routed to the correct UA instance, for example, a call transfer.

Configure SIP Instance Support

Parameter Function	**template** > parameter
Specify if the phone sends sip.instance in the REGISTER request.	reg.x.gruu

Set Local User and Administrator Passwords

The phone prompts you for a user or administrator password before you can access certain menu options. To set the administrator password, change the default administrator password on the phone user interface

in the Settings menu. The default user password is 123 and the default administrative password is 456. 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. If you do not change the default administrative password, the phone displays a reminder message each time the phone reboots.

To remove settings made using the phone's user interface, go to **Reset Local Configuration** menu on the phone.

If the phone requires the administrator password, you may be able to use the user password, but you are 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). If you are registering Polycom phones with Microsoft Skype for Business Server, a message displays on the phone screen prompting you to change the default password.

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 following table for all parameters.

Local User and Administrator Password Settings

Parameter Function	**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
Enable or disable masking of password characters as you type.	**site.cfg** > up.echoPasswordDigits
Set the phone's local administrator password.	**device.cfg** > device.auth.localAdminPassword
Set the phone's local user password.	**device.cfg** > device.auth.localUserPassword

Disable External Ports and Features

You can disable unused external phone ports and phone features to increase the security of devices in your deployment. You can disable the following ports and features:

- **Web Configuration Utility**   A web browser-based interface that enables you to configure settings on the phone.
- **PC port**   Acts as a pass-through switch for externally attached devices such as a personal computer or laptop. The PC port informs the main switch of any secondary (PC) port link status changes.
- **Aux port**   This port is used to connect Polycom Expansion Modules to Polycom VVX phones. This port uses a synchronous peripheral cable interface (SPI) cable.
- **USB Port**   Use this port to plug in devices for local call recording, the picture frame feature, or for the Polycom VVX Camera. VVX 500/501 and 600/601 business media phones have two USB ports. To enable or disable the phone USB port, disable the following three parameters:
  - ➢ `feature.callRecording.enabled=0`
  - ➢ `feature.pictureFrame.enabled=0`
  - ➢ `diags.pcap.enabled=0`

To disable the second USB port on the VVX 500/501 and 600/601, use the parameter:

➢ `video.enable=0`

- **Speakerphone**   Disable the phone's audio speakerphone.
- **Headset**   Disable the phone's headset jack.
- **Handset**   Disable the phone's handset.
- **Call forwarding**   Disable the call forwarding feature.
- **Do Not Disturb**   Disable the Do Not Disturb feature and soft key.
- **Push-to-Talk (PTT)**   Disable the push-to-talk feature.
- **Bluetooth**   Bluetooth capability is disabled by default. To enable or disable Bluetooth headset capability, use the configuration parameter `bluetooth.radioOn`. Note that if `up.headsetmodeenabled` is set to 0 and `bluetooth.radioOn` is set to 1, you can pair a Bluetooth device but no audio is available.
- **Autoanswer menu**   Disable the Autoanswer menu on the phone interface.
- **Applications icon**   Disable the Applications icon on the phone's Home screen.

> **Note: One port must be enabled to send and receive calls**
> At least one audio port must be enabled to send and receive calls.

Disable Unused Ports and Features

Parameter Function	**template** > parameter
Enable or disable the PC port mode that sets the network speed over Ethernet.	**device.cfg** > device.net.etherModePC
Use or do not use all enabled device.xxx fields to set parameters.	**device.cfg** > device.set
Enable or disable the phone auxiliary port.	**device.cfg** > device.auxPort.enable
Enable or disable the complete httpd web client.	**site.cfg** > httpd.enabled
Enable or disable the Web Configuration Utility.	**site.cfg** > httpd.cfg.enabled
Enable or disable push-to talk mode.	**site.cfg** > ptt.pttMode.enable
Enable or disable the phone USB port for local call recording.	**features.cfg** > feature.callRecording.enabled
Enable or disable handsfree mode.	**reg-advanced.cfg** > up.handsfreeMode
Enable or disable the headset port.	**reg-advanced.cfg** > up.headsetModeEnabled
Enable or disable the handset port.	**reg-advanced.cfg** > up.handsetModeEnabled
Enable or disable call forwarding.	**features.cfg** > feature.forward.enable
Turn on or off display of the call forward icon on the phone Home screen.	**features.cfg** > homeScreen.forward.enable
Enable or disable Do Not Disturb (DND).	**features.cfg** > feature.doNotDisturb.enable
Enable or disable display of the DND icon on the phone's Home screen.	**features.cfg** > homeScreen.doNotDisturb.enable

Disable Unused Ports and Features (continued)

Enable or disable the DND soft key on the phone.	**features.cfg** > softkey.feature.doNotDisturb
Enable or disable the phone's Autoanswer menu.	**features.cfg** > call.autoAnswerMenu.enable
Enable or disable the Applicatios icon on the phone's Home screen.	**features.cfg** > homeScreen.application.enable

Example Configuration

This section shows you how to disable external ports and features. You must modify settings on BroadSoft server and for the phones in your deployment. The following example shows the headset, handset, and speakerphone ports disabled.
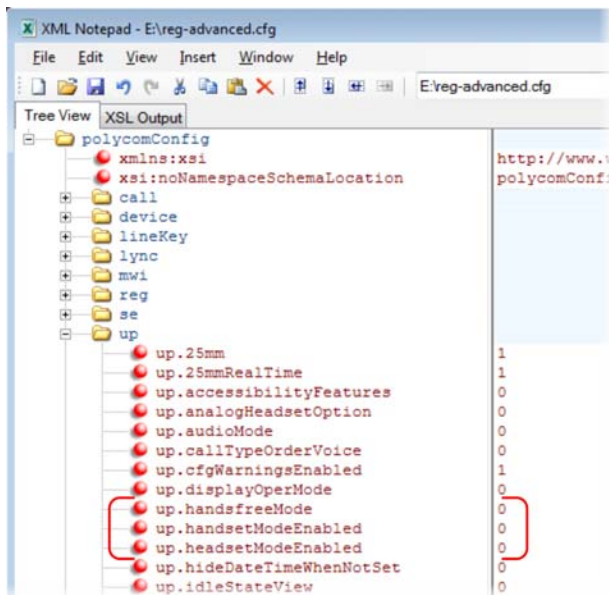
Set Visual Security Classification

This feature enables all phones to display the security classification of an active call. The security classification of a call is determined by the lowest security classification among all participants connected to the call. For example, a Top Secret classification displays when all participants in a call have a Top Secret classification level. If User A is classified as Top Secret and User B has a lower classification level of Restricted, User A and B are connected to the call as Restricted.

> **Caution: Security classifications**
> Calls classification is determined by the lowest classification among all participants in the call. You can safely exchange information classified no higher than the call's security classification.

This feature is supported on Polycom VVX business media phones. To enable this feature, you must configure settings on the BroadSoft BroadWorks server v20 or higher and on the phones. If a phone has multiple registered lines, administrators can assign a different security classification to each line. The following table shows the parameters you can configure on your provisioning server.

An administrator can configure security classifications as names or strings and set the priority of each on the server in addition to the default security classification level Unclassified. The default security classification Unclassified displays until you set classifications on the server. When you establish a call to a phone not connected to this feature, your phone displays as Unclassified.

Note that phone users can modify their assigned security classification level to a value lower than their assigned level during a call, as shown in Modify Security Classification Level. When the call is over, the server resets your classification level to its original state. You cannot change security classification when the phone is in the idle state. You can view the security classification for your phone line by navigating to **Status > Lines.** If your phone has multiple registered lines, you can view the security classification level assigned to each line on the phone in idle state by going to **Home > Settings > Status > Lines > Line x**.

Configure Visual Security Classification

Parameter Function	template > parameter
Enable or disable Visual Security Classification for all lines on a phone.	**sip-interop.cfg** > voIpProt.SIP.serverFeatureControl.securityClassification
Enable or disable Visual Security Classification for a specific phone line.	**reg-advanced.cfg** > reg.x.serverFeatureControl.securityClassification

Modify Security Classification Level

Phone users have the option to modify their classification to a lower classification level at any point during an active call. For example, a user classified as Top Secret can override the classification level to a level lower than Top Secret and revert back to the Top Secret classification at any point during an active call.

To modify a security classification level:

1 Press the **Security** soft key that displays only during an active call.
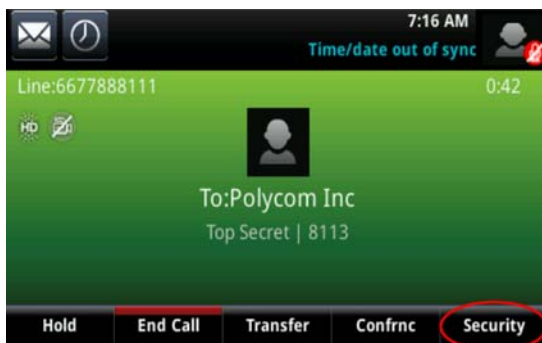
2 In the **Security Classification** screen, choose an available security level and press **Exit**.

In this example, the user changes classification level from **Top Secret** to **Restricted**.
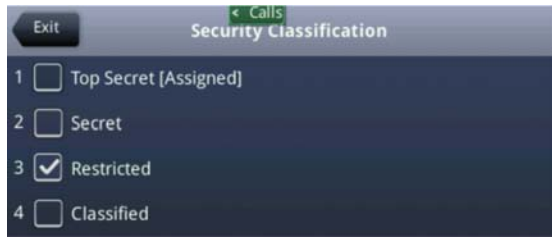
When a participant changes classification level and the change results in a lower classification level for the active call, a message displays to all participants indicating a change in the call's security classification, as illustrated in the following figure.
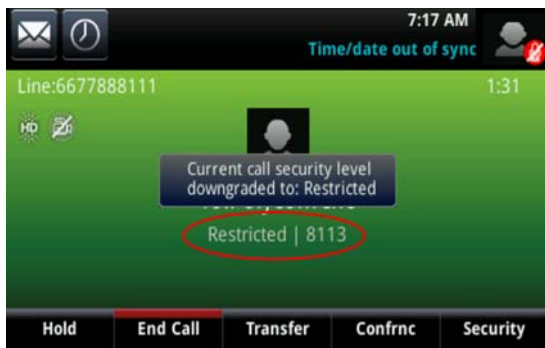
Example Configuration

You must make changes on the BroadSoft r20 server. On your provisioning server, enable the parameter voIpProt.SIP.serverFeatureControl.securityClassification as shown in the following figure.

Enable security classification
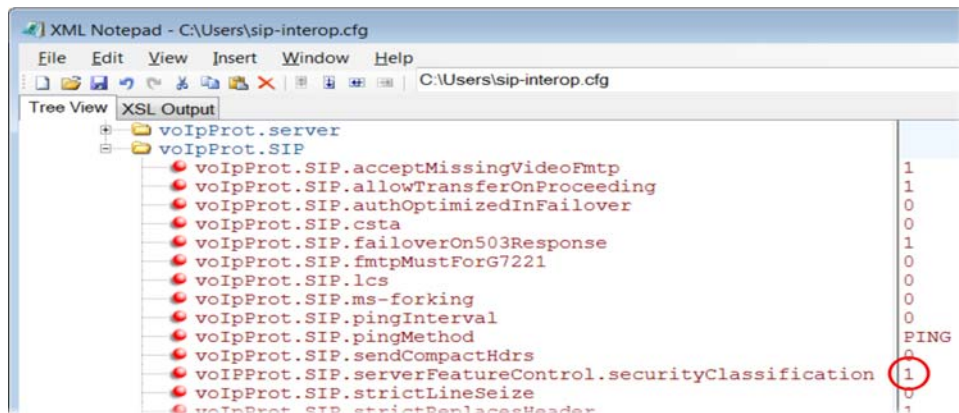
Choose Incoming Signaling Validation

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 following table for the parameters that specify the validation type, method, and the events you want to validate.

Incoming Signal Validation Parameters

Parameter Function	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 is 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

Encrypt Configuration Files

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 following table to configure the parameters used to encrypt files.

Configuration File Encryption Parameters

Parameter Function	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

Use Digital Certificates

Polycom phones are installed with a Polycom-authenticated RSA certificate. You can use this certificate to create a secure connection between phone and server when initiating TLS communications over protocols such as HTTPS and SIP. You can download the Polycom Root CA from http://pki.polycom.com/pki. The certificate is set to expire on March 9, 2044.

 Web Info: Digital certificates on Polycom phones
For details on installing digital credentials on VVX phones, see *Device Certificates on Polycom SoundPoint IP, SoundStation IP, and VVX Phones: Technical Bulletin 37148* at Polycom Engineering ADvisories and Technical Notifications.

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 and indicates that you trust 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.
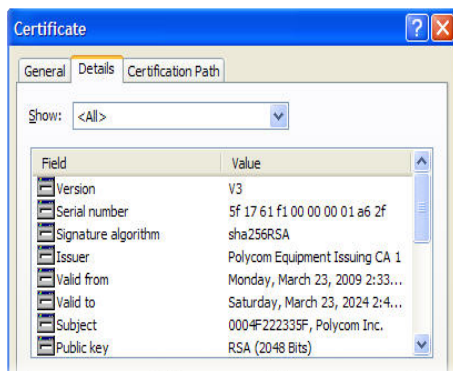
The device certificate and associated private key are stored on the phone in its non-volatile memory as part of the manufacturing process. For more information on digital certificates, see Public Key Infrastructure (X.509) and RFC 2459: Internet X.509 Public Key Infrastructure.

> **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 Polycom phone:

1 Navigate to **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 Press the **Info** soft key to view the certificate.

   One of the following messages display:

   ➢ 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"**
> If your phone reports the device certificate as 'self-signed" rather than "Factory installed," you should return the equipment to receive a replacement.

Generate a Certificate Signing Request

You may need a certificate to perform a number of tasks, for example, multiple TLS authentication. By default, the phone requests a 512-bit certificate. If you require a stronger certificate, use OpenSSL or another certificate signing request utility.

To obtain a certificate you need to:

1 Request a certificate from a Certificate Authority (CA) by creating a certificate signing request (CSR).

2 Forward the CSR to a CA to create a certificate. If your organization doesn't have its own CA, you need to forward the CSR to a company like Symantec.

   If successful, the CA sends back a certificate that has been digitally signed with their private key.

After you receive the certificate, you can download it to the phone in the following ways:

● Using a configuration file

● Through the phone's user interface

● Through the Web Configurable Utility

To generate a certificate signing request on a Polycom phone:

1 Navigate to **Settings > Advanced > Admin Settings > Generate CSR**.

2 When prompted, enter the administrative password and press the **Enter** soft key. The default administrative password is **456**.

3 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 on a VVX 500/501 phone.

4 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 Polycom phones 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 the table TLS Platform Profile and TLS Application Profile Parameters. For the complete list of trusted Certificate Authorities, see the section OpenSSL Versions 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.

After you install certificates on the phone, you can to determine which TLS platform profiles or TLS application profiles 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 to use 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 the section Use Digital Certificates.

> **Web Info: Using custom 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

Parameter Function	**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 Platform Profile and TLS Application Profile Parameters (continued)

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

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

This section includes the following topics:

- Download Certificates to a Polycom Phone
- Set TLS Profiles

Download Certificates to a Polycom Phone

You can download certificates to a Polycom phone 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**
> For VVX 1500 phones, the maximum certificate size on Platform CA1 is 1536KB and 4KB for Platform CA2.

To download a certificate to a Polycom phone:

1   Navigate to **Menu > Settings > Advanced > Administrative Settings > TLS Security** and select **Custom CA Certificates** or **Custom Device Certificates**.

2   When prompted, enter the administrative password and press the **Enter** soft key. The default administrative password is **456**.

3   Select the **Install** soft key.

4   Enter the URL where the certificate is stored.

For example, http://bootserver1.vancouver.polycom.com/ca.crt

5   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.

6   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 following table to refer to 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.

TLS Profile Parameters

Parameter Function	**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.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Support Mutual TLS Authentication

Mutual Transport Layer Security (MTLS) 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. For more information, refer to the section Use Digital Certificates.

Prior to SIP 3.2, and in cases where the phones do not have device certificates, the phone authenticates 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 requests 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.

You can download the Polycom Root CA from http://pki.polycom.com/pki. 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 *Mutual Transport Layer Security Provisioning Using Microsoft Internet Information Services 6.0: Technical Bulletin 52609* at Polycom Engineering Advisories and Technical Notifications.

Configurable TLS Cipher Suites

The phone administrator can control which cipher suites to offer/accept during TLS session negotiation. The phone supports the cipher suites listed in the following table and you can use the parameters listed in the table Configurable TLS Cipher Suites to configure TLS Cipher Suites. The 'Null Cipher' listed in the following table is a special case option which does 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
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 are lost unless you backup the configuration files.

Configurable TLS Cipher Suites

Parameter Function	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

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 advanced encryption standard algorithm. The parameters used to configure SRTP are shown in the following table . 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), a padlock symbol displays in the last frame of the connected context animation (two arrows moving towards each other).

Secure Real Time Transport Protocol Parameters

Parameter Function	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

Secure Real Time Transport Protocol Parameters (continued)

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.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

In example 1, the **srtp_1.cfg** configuration file is shown below:

```
phone
  sec.srtp
    sec.srtp.offer                              1
    sec.srtp.sessionParams.noAuth.offer         1
    sec.srtp.sessionParams.noEncrypRTP.offer    1
    sec.srtp.sessionParams.noEncrypRTCP.offer   1
    sec.srtp.require                            0
    sec.srtp.sessionParams.noAuth.require       0
    sec.srtp.sessionParams.noEncrypRTP.require  0
    sec.srtp.sessionParams.noEncrypRTCP.require 0
```
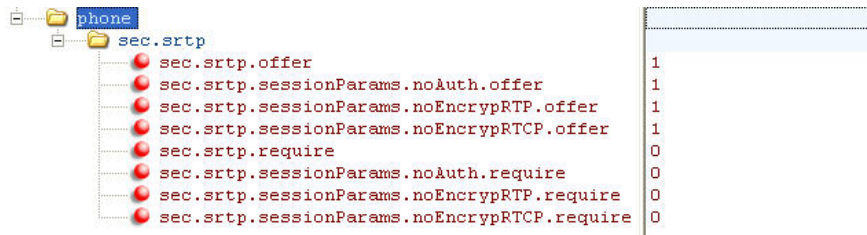
This results in an offer (SIP INVITE with SDP) with 8 crypto attributes with the following session parameters:

```
<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 2, the **srtp_2.cfg** configuration file is shown below:

```
phone
  sec.srtp
    sec.srtp.offer                              1
    sec.srtp.sessionParams.noAuth.offer         1
    sec.srtp.sessionParams.noEncrypRTP.offer    1
    sec.srtp.sessionParams.noEncrypRTCP.offer   1
    sec.srtp.require                            1
    sec.srtp.sessionParams.noAuth.require       0
    sec.srtp.sessionParams.noEncrypRTP.require  1
    sec.srtp.sessionParams.noEncrypRTCP.require 0
```
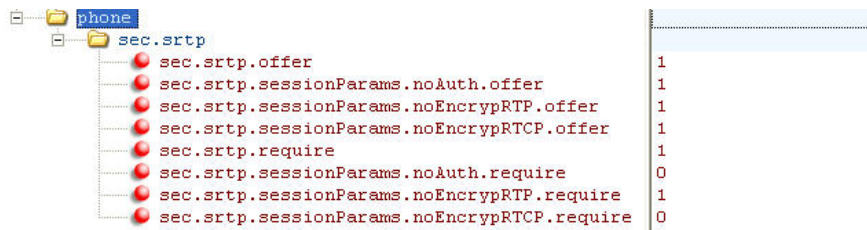
This results in an offer (SIP INVITE with SDP) with 4 crypto attributes with the following session parameters:

```
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

Users can lock their phones and prevent access to the menu or key presses, by pressing 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.

After 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 proceeds. 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 following table 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 displays on the locked phone and the phone's user can answer the call.

Phone Lock Parameters

Parameter Function	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.*

Phone Lock Parameters (continued)

Specify the scenarios when phone lock should be enabled.	**features.cfg** > phoneLock.*

.* indicates grouped parameters. See the section Example Two: Configuring Grouped Parameters for more information.

Secondary Port Link Status Report

Devices equipped with a secondary PC port can act as a pass-through switch for externally attached devices. The phone informs the network switch (authenticator) of any secondary PC port link status changes.

As of Polycom UC Software 3.3.0, Polycom phones include this feature.

A Polycom terminal acting as a pass–through switch

If you want to configure this feature, see the table Secondary Port Link Status Report Parameters for the parameters you need to set. Polycom devices detect an externally connected host connection/disconnection, informing the authenticator switch to initiate the authentication process or drop an existing authentication. This feature ensures that the port authenticated by the externally attached device switches to unauthenticated upon device disconnection so that other unauthorized devices cannot use it. It also ensures that the externally attached device can move to another port in the network and start a new authentication process. This feature extends Cisco Discovery Protocol (CDP) to include a Second Port Status Type, Length, Value (TLV) that informs an authenticator switch of the status of devices connected to a device's secondary PC port.

To reduce the frequency of CDP packets, the phone does not send link up status CDP packets before a certain time period. The phone immediately sends all link-down indication to ensure that the port security is not compromised. You can configure the required elapse time—sleep time—between two CDP UPs dispatching (see **sec.hostmovedetect.cdp.sleepTime**[1]).

If the externally attached device (the host) supports 802.1X authentication, then the device can send an EAPOL-Logoff on behalf of the device after it is disconnected from the secondary PC port. This informs the authenticator switch to drop the authentication on the port corresponding with the previously attached device.

Secondary Port Link Status Report Parameters

Parameter Function	template > parameter
Enable or disable EAPOL logoff.	**site.cfg** > sec.dot1x.eapollogoff.enabled
Specify if the LAN port link should be reset or not.	**site.cfg** > sec.dot1x.eapollogoff.lanlinkreset
Specify the phone should indicate to a host that it has been connected or disconnected to the host's secondary (PC) port.	**site.cfg** > sec.hostmovedetect.cdp.enabled
Set the time interval between link-up and link-down reporting.	**site.cfg** > sec.hostmovedetect.cdp.sleepTime

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). IEEE 802.1X is based on the Extensible Authentication Protocol (EAP). As of Polycom UC Software 3.3.0, Polycom phones support standard IEEE 802.1X authentication. The following figure shows a typical 802.1X network configuration with wired Polycom phones.

A typical 802.1X network configuration
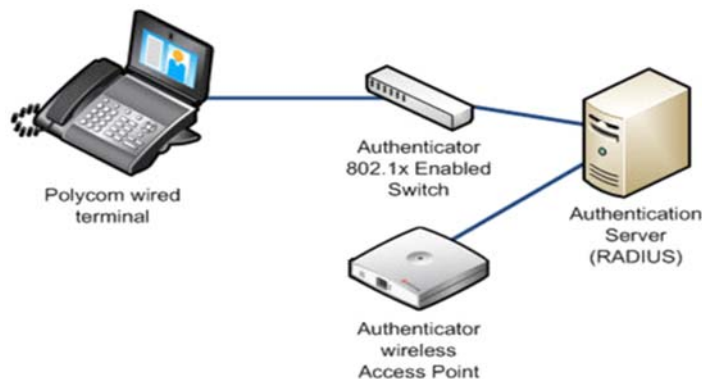
Polycom phones support the following EAP authentication methods:

- 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 following table S to configure 802.1X Authentication. For more information see Configure TLS Profiles.

> **Web Info: EAP authentication protocol**
> For more information, see RFC 3748: Extensible Authentication Protocol.

Set 802.1X Authentication Parameters

Parameter Function	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

Set User Profiles

You can allow users to access their personal phone settings from any phone in the organization. This feature is particularly useful for remote and mobile workers who do not have a dedicated work space and conduct business in more than one location. The user profile feature is also useful if an office has a common conference phone from which multiple users need to access their personal settings.

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**. After a user logs in, the user can:

- Access personal phone settings such as contacts, speed dials, call lists
- Change user password
- Log out of a phone

If a user changes any settings while logged in to a phone, the settings save and display 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.

> **Tip: Calling authorized and emergency numbers from shared devices**
> 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.

You must decide whether to require users to always log in to a phone or not. If you set up the user profile feature as enabled, and users are not required to log in, users have the option to use the phone as is—without access to their personal settings—or they can log in to display their personal settings. You can also specify if a user is logged out of the phone when the phone restarts or reboots.

You can choose to define default credentials for a default, generic phone account (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 generic phone profile displays, and users retain 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 in the format **<user>.cfg** to specify 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 causes 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.

Server Authentication of User Profiles

Instead of phone-based authentication of user profiles, you can configure server authentication. When you enable server authentication, you set up user accounts on the provisioning server and each user authenticates by entering correct server credentials. This feature is available on all VVX business media phones.

To use this feature, you must create a generic profile and generic credentials on the provisioning server. Next, in the Home directory of each user, create a user profile with a user-specific configuration file that you store on the provisioning server with a unique name, for example <username>.cfg, as well as user-specific files such as settings, directory, and call lists. When a user logs in with credentials, the phone downloads the user profile from the provisioning server. When the user logs out, the phone downloads the default user profile using the generic credentials.

Note that if an unexpected reboot occurs while a user is logged in, the user is not logged out and the phone returns to the user profile after reboot.

The phone downloads log files app.log and boot.log from the generic profile on the provisioning server regardless of user logins.

If a user forgets to log out and prevents other users from logging in, the user can ask the administrator to log out remotely. Administrators can log out a user remotely with a checksync event in the NOTIFY by setting the parameter `profileLogout=remote`.

Create a Generic Profile Using Server Authentication

Create a generic profile for the phones when no user is logged in.

If you enable server authentication of user profiles, the following parameters do not apply and you do not need to configure them:

- `prov.login.defaultUser`
- `prov.login.defaultPassword`
- `prov.login.defaultOnly`
- `prov.login.localPassword`
- `prov.login.localPassword.hashed`

To create a generic profile:

1 On the server, create an account and directory for the generic profile, for example, '*Generic_Profile*'.

2 In the Generic directory, create a configuration file for a generic profile the phone uses by default, for example, *<genericprofile>*.cfg.

3 In *<genericprofile>*.cfg, include registration and server details and set all phone feature parameters. You must set the following parameters to use server authentication:

  ➢ `prov.login.enabled="1"`

  ➢ `prov.login.useProvAuth="1"`

  ➢ `prov.login.persistent="1"`

  Note that if you enable `prov.login.enabled=1` and do not enable `prov.login.useProvAuth=0`, users are authenticated by a match with credentials you store in the user configuration file *<user>*.cfg.

4 Create a master configuration file 000000000000.cfg for all the phones, or a *<MACAddress>*.cfg for each phone, and add *<genericprofile>*.cfg to the CONFIG_FILES field.

  For information about using the master configuration file, see 'Provision and Configure Phones with Polycom UC Software' in the UC Software Administrator's Guide on Polycom Voice Support.

5 Set the user name and password credentials on the phone at **Settings > Advanced > Provisioning Server** details and inform users of their user profile credentials.

Note that the following override files are uploaded to the generic profile directory:

- Log files
- Phone menu settings
- Web Configuration Utility settings
- Call logs
- Contact directory file

Create a User Profile Using Server Authentication

Create a user profile when using server authentication.

To create a user profile:

1 On the server, create an account and a directory for each user, for example, '*User_1*', '*User_2*"…

2 In each user directory, create a configuration file for each user, for example, *<User_1>*.cfg, *<User_2>*.cfg, that contains the user's registration details and feature settings.

The following override files are uploaded to the generic profile account on the server:

- Log files
- Web Configuration Utility settings

The following override files are uploaded to the user profile account on the server:

- Phone menu settings
- Call logs
- Contact directory file

User Profile Parameters

Parameter template	Permitted Values
prov.login.automaticLogout site.cfg	Specify the amount of time before a non-default user is logged out. 0 minutes (default) 0 to 46000 minutes
prov.login.defaultOnly site.cfg	0 (default) - The phone cannot have users other than the default user. 1 - The phone can have users other than the default user.
prov.login.defaultPassword site.cfg	Specify the default password for the default user. NULL (default)
prov.login.defaultUser site.cfg	Specify the name of the default user. If a value is present, the user is automatically logged in when the phone boots up and after another user logs out. NULL (default)
prov.login.enabled site.cfg	0 (default) - The user profile is disabled. 1 - The user profile feature is enabled.
prov.login.localPassword.hashed site.cfg	0 (default) - The user's local password is formatted and validated as cleartext. 1 - The user's local password is created and validated as a hashed value.
prov.login.localPassword site.cfg	Specify the password used to validate the user login. The password is stored either as plain text or as an encrypted SHA1 hash. 123 (default)
prov.login.persistent site.cfg	0 (default) - Users are logged out if the handset reboots. 1 - Users remain logged in when the phone reboots.
prov.login.required site.cfg	0 (default) - The user does not have to log in. 1 - The user must log in when the login feature is enabled.
prov.login.useProvAuth site.cfg	0 (default) - The phone do not user server authentication. 1 - The phones use server authentication and user login credentials are used as provisioning server credentials.
voIpProt.SIP.specialEvent.checkSync.downloadCallList site.cfg	0 (default) - The phone does not download the call list for the user after receiving a checksync event in the NOTIFY. 1 - The phone downloads the call list for the user after receiving a checksync event in the NOTIFY.

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 contains 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 specifies 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 to log in to the phone. The name of the file is 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 displays 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 reference section Configuration Parameters.

The following is a sample user configuration file.

```
polycomConfig
    xmlns:xsi                              http://www.w3.org/2001/XMLSchema-instance
    xsi:noNamespaceSchemaLocation          polycomConfigPrivate.xsd
    #comment                               User Profile
    prov.login
        prov.login.localPassword           123
        prov.login.localPassword
            prov.login.localPassword.hashed  0
    #comment                               Registration definition
    reg
    #comment                               Sampled audio definition
    saf
    #comment                               Feature definition
    feature
```

If a user updates their password or other user-specific settings using the Main Menu on the phone, the updates are stored in **<user>-phone.cfg**, not **<MACaddress>-phone.cfg**.

If a user updates their contact directory while logged in to a phone, the updates are stored in **<user>-directory.xml**. Directory updates display each time the user logs in to a phone. For certain phones (for example, the VVX 1500 phone), an up-to-date call lists history is defined in **<user>-calls.xml**. This list is 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)
- Configuration files listed in the master configuration file (including <user>.cfg)
- Default values

Configure I-Frames

When video streams initialize, devices transmit video packets called I-frames (reference frames) that contain information to display a complete picture. Subsequent video packets, known as P-frames, are smaller and not as complete to consume less bandwidth. Due to packet loss, jitter, or corruption, devices occasionally need to make multiple requests for a complete I-frame in order to reset the full frame, after which devices can revert to P-frame updates.

You can set parameters to control an I-frame request. The following table indicates parameter dependencies and messaging behavior when setting an I-frame request method.

I-Frame Parameter Dependencies

video. forceRtcpVideoCodec Control	video. dynamicControlMe thod	voIpProt. SDP.offer.rtcpVideoCodec Control	Behavior when requesting video I-frame updates
0	0 (n/a)	0	Only SIP INFO messages are sent. No RTCP-FB is offered in SDP.
0	1 (n/a)	0	Only SIP INFO messages are sent. No RTCP-FB is offered in SDP.
0	0 (n/a)	1	RTCP-FB is offered in SDP. If SDP responses do not contain the required RTCP-FB attribute, then only SIP INFO requests are used.
0	1 (n/a)	1	RTCP-FB is offered in SDP. If SDP responses do not contain the required RTCP-FB attribute, then only SIP INFO requests are used.
1	0	0	The SDP attribute a=rtcp-fb is not included in SDP offers. Both RTCP-FB and SIP INFO messages are attempted.
1	1	0	The SDP attribute a=rtcp-fb is not included in SDP offers. Both RTCP-FB and SIP INFO messages are attempted. If no RTCP-FB messages are received, only SIP INFO messages are sent. If no response is received for SIP INFO messages then, again, both RTCP-FB and SIP INFO messages are attempted.

I-Frame Parameter Dependencies (continued)

video. forceRtcpVideoCodec Control	video. dynamicControlMe thod	voIpProt. SDP.offer.rtcpVideoCodec Control	Behavior when requesting video I-frame updates
1	0	1	RTCP-FB is offered in SDP. Even if the SDP response does not include an accepted a=rtcp-fb attribute both RTCP-FB and SIP INFO messages are sent.
1	1	1	RTCP-FB is offered in SDP. Even if the SDP response does not include an accepted a=rtcp-fb attribute both RTCP-FB and SIP INFO messages are sent initially. If no RTCP-FB response is received, only SIP INFO messages are sent afterwards.

Configuration Parameters

This section is a reference guide to the UC Software configuration parameters you use to configure devices and call controls. This section provides a description of each configuration parameter, and permitted and default values.

The following table shows parameters for the SIP-B automatic call distribution (ACD) and feature synchronized ACD features.

Automatic Call Distribution Parameters

Parameter	Permitted Values	Default
acd.reg[1]	**1 to 34**	**1**
The index of the registration (line) used to support BroadSoft server-based ACD.		
acd.stateAtSignIn	**0 or 1**	**1**
The state of the user when signing in. If 1, the user is available. If 0, the user is unavailable.		
acd.x.unavailreason.active	**0 or 1**	**0**
If 1, the reason code is active. If 0, the code is inactive.		
acd.x.unavailreason.codeValue[1]	String	Null
The code value. For example, 1000100000		
acd.x.unavailreason.codeName[1]	string	Null
The code name. For example, Out to Lunch		
These three parameters configure the unavailable reason codes used for premium feature-synchronized ACD features, where x is the index of up to 100 codes.		

[1] Change causes phone to restart or reboot.

The following table lists parameters you can use to control telephone notification events, state polling events, and push server controls.

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**

Application Parameters (continued)

Parameter	Permitted Values	Default
Choose a priority level for push messages from the application server to the phone. **1** (None) Discard push messages **2** (Normal) Allows only normal push messages **3** (Important) Allows only important push messages **4** (High) Allows only priority push messages **5** (Critical) Allows only critical push **6** (All) Allows all push messages		
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**	**1**
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 sent to the microbrowser is `http://172.24.128.85:8080/sampleapps/examples/sample.html.` You can use HTTP or HTTPS.		
apps.push.username	**string**	**null**
The user name to access the push server URL. Note: To enable the push functionality, you must set values for the parameters `apps.push.username` and `apps.push.password` (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.		

Application Parameters (continued)

Parameter	Permitted Values	Default
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.		
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.URL	**URL**	**null**
The URL to which the phone sends notifications of specified events. You can use 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. You can use 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.orientation	**Unspecified, Left, Right**	**Unspecified**
The location of the VVX 500/501 and 1500 with respect to the user's computer. For example, to the `Left` of the computer.		
apps.ucdesktop.ServerAddress	**string**	**null**
The user's computer as a fully qualified domain name (FQDN), for example, computer@yourcompany.com.		

Application Parameters (continued)

Parameter	Permitted Values	Default
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.

The busy lamp field (BLF)/attendant console feature enhances support for phone-based monitoring. The maximum number of BLF entries for these phones is 50. The parameters listed in the following table are supported on the VVX 300 series, 400 series, 500 series, and 600 series phones.

In the following table, x in a parameter is the number of the BLF entry in the list. If you are using static BLF, you need to configure the number of each entry.

Attendant/Busy Lamp Field Parameters

Parameter	Permitted Values	Default
attendant.reg	**positive integer**	**1**
The index of the registration to use to send a SUBSCRIBE to the list SIP URI specified in `attendant.uri`. For example, `attendant.reg = 2` means the second registration is used.		
attendant.ringType	**default, ringer1 to ringer24**	**ringer1**
The ringtone to play when a BLF dialog is in the offering state.		
attendant.uri	**string**	**Null**
The list SIP URI on the server. If this is just a user part, the URI is constructed with the server hostname/IP.		
Note: If this parameter is set, then the individually addressed users configured by `attendant.resourceList` and `attendant.behaviors` are ignored.		
attendant.behaviors.display.spontaneousCallAppearances.normal **Normal**	**0 or 1**	**1**
attendant.behaviors.display.spontaneousCallAppearances.automata **Automatic**	**0 or 1**	**0**
If 1, the normal or automatic call appearance is spontaneously presented to the attendant when calls are alerting on a monitored resource (and a ring tone is played). If 0, the call appearance is not spontaneously presented to the attendant. The information displayed after a press and hold of a resource's line key is unchanged by this parameter. Note that the values of these call appearance parameters depend on the values applied to `attendant.resourceList.x.type`.		
attendant.behaviors.display.remoteCallerID.normal **Normal** **attendant.behaviors.display.remoteCallerID.automata** **Automatic**	**0 or 1**	**1**

Attendant/Busy Lamp Field Parameters (continued)

Parameter	Permitted Values	Default
These parameters depend on the value set for the parameter `attendant.resourceList.x.type`. If the parameter `attendant.resourceList.x.type` is set to normal, use the parameter `attendant.behaviors.display.remoteCallerID.normal`. If the parameter `attendant.resourceList.x.type` is set to automata, use the parameter `attendant.behaviors.display.remoteCallerID.automata`. If 1, normal and automatic remote party caller ID information is presented to the attendant. If 0, the string `unknown` is substituted for both name and number information.		
attendant.resourceList.x.address	**string that constitutes a valid SIP URI (sip:6416@p ol ycom.com) or contains the user part of a SIP URI (6416)**	**Null**
The user referenced by `attendant.reg=""` subscribes to this URI for dialog. If a user part is present, the phone subscribes to a sip URI constructed from the user part and domain of the user referenced by `attendant.reg`.		
attendant.resourceList.x.bargeInMode	**All, Normal, Listen, Whisper**	
Enable or disable barge -in and choose the default barge-in mode. Note: The default value for this parameter is empty. If no value is entered, the Barge In feature is disabled.		
attendant.resourceList.x.callAddress	**string**	**Null**
If the BLF call server is not at the same address as the BLF presence server, calls are sent to this address instead of the address specified by `attendant.resourceList.x.address`.		
attendant.resourceList.x.label	**UTF-8 encoded string**	**Null**
The text label displays adjacent to the associated line key. If set to Null, the label is derived from the user part of `attendant.resourceList.x.address`.		
attendant.resourceList.x.proceedingIsRecipient	**0 or 1**	**0**
A flag to determine if pressing the associated line key for the monitored user picks up the call.		
attendant.resourceList.x.requestSilentBargeIn	**0 or 1**	**0**
Enable or disable a tone that plays when a contact barges in on a call. If 0, a tone plays when a contact barges in on a call. If 1, no tone is played when a contact barges in on a call.		

Attendant/Busy Lamp Field Parameters (continued)

Parameter	Permitted Values	Default
attendant.resourceList.x.type	**normal or automata**	**normal**

The type of resource being monitored and the default action to perform when pressing the line key adjacent to monitored user x.

If `normal`, the default action is to initiate a call if the user is idle or busy and to perform a directed call pickup if the user is ringing. Any active calls are first placed on hold. Note that the value `normal` applies the call appearance setting `attendant.behaviors.display.*.normal`.

If `automata`, the default action is to perform a park/blind transfer of any currently active call. If there is no active call and the monitored user is ringing/busy, an attempt to perform a directed call pickup/park retrieval is made. Note that the value `automata` applies the call appearance setting `attendant.behaviors.display.*.automata=0`.

[1] Change causes phone to restart or reboot.

Use this parameter to toggle between audio-only or audio-video calls.

Audio Video Persist Parameters

Parameter	*Permitted Values*	*Default*
audioVideoToggle.callMode.persistent	**0 or 1**	**0**

If 0, the phone returns to audio-only for outgoing calls. If 1, the last call mode the user selected persists until you change modes using the soft key toggle. Note that you must enable `feature.audioVideoToggle.enabled="1"` to apply this parameter.

The parameters listed in the following table control how you display background images.

Background Parameters

Parameter	Permitted Values	Default
bg.background.enabled	**0 or 1**	**1**

Enable or disable the background image change feature. When set to 1, the user can set the background image on the phone screen. When set to 0, the user cannot set the background image of the phone screen and:

The background image option is not available on the phone menu or in the Web Configuration Utility when logged in as a user.

The icon to set the displayed image as a background in the picture frame menu does not display.

Administrators can change the background image using a configuration file or by logging into the Web Configuration Utility as an administrator.

bg.color.selection	**w,x**	**1,1**

Background Parameters (continued)

Parameter	Permitted Values	Default
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 4 to select one of the four `solid` backgrounds. Use w=3 and x= 1 to 6 to select one of the six background `bm` images You can set backgrounds for specific phone models by adding the model name, for example: `bg.color.VVX500.selection, bg.color.VVX1500.selection` Note that although the VVX 300 series phones use a grayscale background, use this parameter to set the background.		
bg.color.bm.x.name **Phone screen background image file**	URL or file path of a BMP or JPEG image	
bg.color.bm.x.em.name **Expansion module (EM) background image file**	URL or file path of a BMP or JPEG image	
The name of the image file (including extension). Note: If the file is missing or unavailable, the built-in default solid pattern is displayed.		

The following table specifies the Bluetooth parameter for the VVX 600/601 phone.

Bluetooth Radio Transmitter Parameter

Parameter	Permitted Values	Default
bluetooth.pairedDeviceMemorySize	**0 – 10**	**10**

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 cause the missed-calls counter to increment.

You can enable/disable missed call tracking on a per-line basis.

In the following table, x is the registration number. To view the list of maximum registrations for each phone model see the table Flexible Call Appearances.

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.H323	**0 or 1**	**0**

You can use this parameter for the VVX 500/501, 600/601, and 1500. If 0, auto-answer is disabled for H.323 calls. If 1, auto-answer is enabled for all H.323 calls.

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 are overridden such that a ringtone of `visual` (no ringer) applies.

call.autoAnswer.SIP	**0 or 1**	**0**

You can use this parameter on the VVX 300 series, 400 series, 500 series, 600 series, and 1500. If 0, auto-answer is disabled for SIP calls. If 1, auto-answer is enabled for all SIP calls.

call.autoAnswer.videoMute	**0 or 1**	**0**

You can use this parameter for the VVX 500/501, 600/601, and 1500. If 0, video begins transmitting (video Tx) immediately after a call is auto-answered. If 1, video transmission (video Tx) is initially disabled after a call is auto-answered.

call.autoAnswer.enable	**0 or 1**	**1**

If 1, the autoanswer menu displays and is available to the user to configure. If 0, the autoanswer menu is disabled and is not available to the user to configure.

Call Parameters (continued)

Parameter	Permitted Values	Default
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 or H323**	**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.

Only the VVX 500/501, 600/601, and 1500 phones use the `protocol` parameter. If no protocol is specified, the phone uses the protocol specified by `call.autoRouting.preferredProtocol`. If a line is configured for a single protocol, the configured protocol is used.

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).

call.autoRouting.preferredProtocol	**SIP or H323**	**SIP**

You can use this parameter for the VVX 500/501, 600/601, and 1500. If set to **SIP**, calls are placed via SIP if available, or via H.323 if SIP is not available. If set to **H323**, calls are placed via H.323 if available, or via SIP if H.323 is not available.

call.autoRouting.preference	**line or protocol**	**line**

You can use this parameter for the VVX 500/501, 600/601, and 1500. If set to **line**, calls are placed via the first available line, regardless of its protocol capabilities. If the first available line has both SIP and H.323 capabilities, the preferred protocol is used (`call.autoRouting.preferredProtocol`). If set to **protocol**, the first available line with the preferred protocol activated is used, if available. If not available, the first available line is used. Note: Auto-routing is used when manual routing selection features (`up.manualProtocolRouting`) are disabled.

call.BlindTransferSpecialInterop	**0 or 1**	**0**

Set the value to 1 to wait for an acknowledgment from the transferee before ending the call.

call.callsPerLineKey	**Varies by phone model**	**Varies by phone model**

Set the maximum number of concurrent calls per line key. This parameter applies to all registered lines. The maximum number of concurrent calls per line key varies by phone model and is listed for each phone in the column *Calls Per Line Key* in the table Flexible Call Appearances. For more information on all types of call appearances see the section Call Forward on Shared Lines. 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 Parameters (continued)

Parameter	Permitted Values	Default
call.defaultTransferType	**Consultative or Blind**	**Generic SIP = Consultative** **Lync = Blind**

Set the transfer type the phone uses when transferring a call. If Blind, pressing the Transfer soft key immediately transfers the call to another party. If Consultative, pressing the Transfer soft key puts the call on hold while placing a new call to the other party.

The user can press and hold the Transfer soft key to change the transfer type temporarily. The user can also set the default transfer type by going to **Settings > Basic > Preferences > Default Transfer Type**.

call.dialtoneTimeOut[1]	**positive integer**	**60**

The time is seconds that a dial tone plays before a call is dropped. If set to 0, the call is not dropped.

call.directedCallPickupMethod[1]	**native or legacy**	**legacy**

Specifies how the phone performs a directed call pick-up from a BLF contact.
- **native**   Indicates the phone uses a native protocol method (in this case SIP INVITE with the Replaces header).
- **legacy**   Indicates the phone uses the method specified in `call.directedCallPickupString`.

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 applies to all registrations on the phone (globally), or apply on a per-registration basis. If 0, DND applies 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. Setting this parameter to 1 allows Polycom VVX 500/501, 600/601, and 1500 phones to make calls using the H.323 protocol even though an H.323 gatekeeper is not configured.

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.hold.remoteNotification.enabled	**0 or 1**	**0**

Call Parameters (continued)

Parameter	Permitted Values	Default
call.hold.remoteNotification.period	**1 - 3600**	**60**
call.hold.remoteNotification.startDelay	**1 - 3600**	**60**
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.

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.localConferenceEnabled[1]	**0 or 1**	**1**

If set to 1, on the VVX 300 series, 400 series, 500 series, and 600 series, the Conference and Join soft keys display during an active call and you can establish conferences on the phone.

If set to 0, on the VVX 300 series, 400 series, 500 series, and 600 series, the Conference and Join soft keys do not display during an active call.

If set to 0, and you press the Conference hard key on the VVX 1500, an 'Unavailable' message displays.

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 missed call 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 Menu > 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 missed calls 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 missed calls depends on how the server is configured.

call.offeringTimeOut[1]	**positive integer**	**60**

Specify a time in seconds that an incoming call rings before the call is dropped, 0=infinite.

Note**:** The call diversion, no answer feature takes precedence over this feature if enabled.

call.parkedCallRetrieveMethod[1]	**native or legacy**	**Null**

The method the phone uses to retrieve a BLF resource's call which has dialog state confirmed. **native** indicates the phone uses a native protocol method (in this case SIP INVITE with the Replaces header). **legacy** indicates the phone uses the method specified in `call.parkedCallRetrieveString`.

Call Parameters (continued)

Parameter	Permitted Values	Default
call.parkedCallRetrieveString[1]	**star code**	***88**
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 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.		
call.shared.exposeAutoHolds[1]	**0 or 1**	**0**
If 1, a re-INVITE is sent to the server when setting up a conference on a shared line. If 0, no re-INVITE is 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. A quick press and release of the line key resumes a call whereas pressing and holding down the line key shows a list of calls on that line.		
call.shared.preferCallInfoCID	**0 or 1**	**0**
Specified whether Caller ID information is displayed. 0 (default) - Caller ID received from 200OK is ignored if NOTIFY message includes display information. 1 - Caller ID received from 200OK is displayed if NOTIFY message includes display information.		
call.shared.remoteActiveHoldAsActive	**0 or 1**	**1**
If 1, shared remote active/hold calls are treated as a active call on the phone. If 0, shared remote active/hold calls are not treated as a active call on the phone.		
call.shared.seizeFailReorder[1]	**0 or 1**	**1**
If set to 1, play re-order tone locally on shared line seize failure.		

Call Parameters (continued)

Parameter	Permitted Values	Default
call.singleKeyPressConference	**0 or 1**	**0**

If set to 1, a conference is initiated when 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 heard only by the conference initiator.

Parameter	Permitted Values	Default
call.stickyAutoLineSeize[1]	**0 or 1**	**0**

If set to 1, the phone uses sticky line seize behavior. This helps with features that need a second call object to work with. The phone attempts 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 uses the line index for the previous call. Dialing through the call list when there is an active call uses the current active call line index. Dialing through the contact directory uses 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 uses the line index for the previous call. Dialing through the contact directory uses 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.

Parameter	Permitted Values	Default
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.

Parameter	Permitted Values	Default
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).

Parameter	Permitted Values	Default
call.urlModeDialing[1]	**0 or 1**	**0**

If 0, URL dialing is disabled. If 1, URL dialing is enabled.

Parameter	Permitted Values	Default
call.urlNumberModeToggling	**0 or 1**	**0**

Determines whether the phone uses Number mode or URL mode when a URL call is initiated.

0 (default) - URL mode is used for URL calls.

1 - Number mode is used for URL calls.

[1] Change causes phone to restart or reboot.

The call lists (or call log) parameters listed in the following table are supported on VVX 300 series, 400 series, 500 series, 600 series, and 1500 phones.

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 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.		

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 Product Shipping Configuration Change Notice at Polycom Engineering Advisories and Technical Notifications.

Polycom provides a global `device.set` parameter that you must enable to install software and change device parameters. In addition, each `<device/>` parameter has a corresponding `.set` parameter that enables or disables the value for that device parameter. You need to enable the corresponding `.set` parameter for each parameter you want to apply.

After you complete the software installation or configuration changes to device parameters, remove `device.set` to prevent the phones from rebooting and triggering a reset of device parameters that phone users might have changed after the initial installation.

If you configure any parameter values using the <device/> parameters, any subsequent configuration changes you make from the Web Configuration Utility or phone local interface do not take effect after a phone reboot or restart.

The `<device/>` parameters are designed to be stored in flash memory, and are therefore not added to the `<MAC>-web.cfg` or `<MAC>-phone.cfg` override files whether the changes are made through the web interface or the phone interface. This design protects the ability to manage and access the phones using the standard set of parameters on a provisioning server after the initial installation.

.set Parameter Exception

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 `device.sec.TLS.customDeviceCertX.publicCert` and to `device.sec.TLS.customDeviceCertX.privateKey`.

> **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 `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 displays 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 address conflicts, Polycom recommends that you test the new configuration files on two phones before initializing all phones.

Types of Device Parameters

The following table 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 Parameter Types (continued)

device.xxx[1]	string	

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`.

[1] Change causes phone to restart or reboot

The following table lists each of the <device/> parameters that you can configure.

Device Parameters

Parameter	Permitted Values	Default
device.auth.localAdminPassword	**string (32 character max)**	

The phone's local administrative password. The minimum length is defined by `sec.pwd.length.admin<XREF>`.

device.auth.localUserPassword	**string (32 character max)**	

The phone user's local password. The minimum length is defined by `sec.pwd.length.user<XREF>`.

device.auxPort.enable[1]	**0 or 1**	**1**

Enable or disable the phone auxiliary port.

device.baseProfile	**Generic, Lync**	

Choose the Base Profile that the phone operates with.

device.dhcp.bootSrvOpt[1]	**Null, 128 to 254**	

When the boot server is set to Custom or Custom+Option66, specify the numeric DHCP option that the phone looks for.

device.dhcp.bootSrvOptType[1]	**IP address or string**	

The type of DHCP option the phone looks 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`.

device.dhcp.bootSrvUseOpt[1]	**Default, Custom, Static, CustomAndDefault**	

Default   The phone looks 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 looks 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 uses the boot server configured through the provisioning server `device.prov.*` parameters.

Custom and **Default**   The phone uses the custom option first or use Option 66 if the custom option is not present.

Device Parameters (continued)

Parameter	Permitted Values	Default
device.dhcp.enabled[1]	**0 or 1**	
If 0, DHCP is disabled. If 1, DHCP is enabled.		
device.dhcp.option60Type[1]	**Binary, ASCII**	
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**	
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` is ignored). `Custom`, use the number specified by `device.dhcp.dhcpVlanDiscOpt`.		
device.dhcp.dhcpVlanDiscOpt[1]	**128 to 254**	
The DHCP private option to use when `device.dhcp.dhcpVlanDiscUseOpt` is set to `Custom`.		
device.dns.altSrvAddress[1]	**server address**	
The secondary server to which the phone directs domain name system (DNS) queries.		
device.dns.domain[1]	**string**	
The phone's DNS domain.		
device.dns.serverAddress[1]	**string**	
The primary server to which the phone directs DNS queries.		
device.host.hostname[1]	**string**	
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.net.cdpEnabled[1]	**0 or 1**	
If set to 1, the phone attempts to determine its VLAN ID and negotiate power through CDP.		
device.net.dot1x.anonid[1]	**string**	
EAP-TTLS and EAP-FAST only. The anonymous identity (user name) for 802.1X authentication.		
device.net.dot1x.enabled[1]	**0 or 1**	
If 0, 802.1X authentication is disabled. If 1, 802.1X authentication is enabled.		
device.net.dot1x.identity[1]	**string**	
The identity (user name) for 802.1X authentication.		

Device Parameters (continued)

Parameter	Permitted Values	Default
device.net.dot1x.method	**EAP-None, EAP-TLS, EAP-PEAPv0-MSCHAPv 2, EAP-PEAPv0-GTC, EAP-TTLS-MSCHAPv2, EAP-TTLS-GTC, EAP-FAST, EAP-MD5**	
Specify the 802.1X authentication method, where EAP-NONE means no authentication.		
device.net.dot1x.password[1]	**string**	
The password for 802.1X authentication. This parameter is required for all methods except EAP-TLS.		
device.net.etherModeLAN[1]	**Auto, 10HD, 10FD, 100HD, 100FD, 1000FD**	
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, 1000FD**	**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**	
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.etherVlanFilter[1]	**0 or 1**	
VLAN filtering for VVX phones is done by the Linux operating system and it cannot be disabled.		
device.net.ipAddress[1]	**string**	
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]	**IP address**	
The phone's default router.		
device.net.lldpEnabled[1]	**0 or 1**	
If set to 1, the phone attempts to determine its VLAN ID and negotiate power through LLDP.		
device.net.subnetMask[1]	**subnet mask**	
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**	
The phone's 802.1Q VLAN identifier. If Null, no VLAN tagging.		

Device Parameters (continued)

Parameter	Permitted Values	Default
device.prov.maxRedunServers[1]	**1 to 8**	
The maximum number of IP addresses to use from the DNS.		
device.prov.password[1]	**string**	
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 re-provisions. The phone may also reboot if the configuration on the provisioning server has changed.		
device.prov.redunAttemptLimit[1]	**1 to 10**	
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**	
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	**IP address, domain name string, or URL**	
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 re-provisions. The phone may also reboot if the configuration on the provisioning server has changed.		
device.prov.serverType[1]	**FTP, TFTP, HTTP, HTTPS, FTPS**	
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 0 -255 characters**	**NULL**
Specify the URL or path for a software version to download to the device. On the Web Configuration Utility, the path to the software version you specify displays in the drop-down list at Utilities > Software Upgrade > Check for Updates. On the phone, enter the path to the software version at Settings > Advanced > Administration Settings > Network Configuration > Provisioning Server > Upgrade Server.		
device.prov.tagSerialNo	**0 or 1**	
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**	
The user name required for the phone to log in to the provisioning server (if required). Note: If you modify this parameter, the phone re-provisions. The phone may also reboot if the configuration on the provisioning server has changed.		
device.prov.ztpEnabled	**0 or 1**	

Device Parameters (continued)

Parameter	Permitted Values	Default
If 0, Disable the ZTP feature. If 1, enable the ZTP feature. For information, see Polycom Zero Touch Provisioning Solution.		
device.sec.configEncryption.key[1]	**string**	
The configuration encryption key used to encrypt configuration files. For more information, see the section Encrypt Configuration Files.		
device.sec.coreDumpEncryption.enabled	**0 or 1**	**1**
This parameter enables you to bypass the encryption of the core dump. When set to 1, the core dump is encrypted. When set to 0, encryption of the core dump is bypassed.		
device.sec.TLS.customCaCert1 (TLS Platform Profile 1) **device.sec.TLS.customCaCert2 (TLS Platform Profile 2)**	**string, PEM format**	
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)**	
device.sec.TLS.customDeviceCert1.privateKey **device.sec.TLS.customDeviceCert2.privateKey**	**Enter the corresponding signed private key in PEM format (X.509)**	
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 enables or disables 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**	
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 Parameters (continued)

Parameter	Permitted Values	Default
device.sec.TLS.profile.cipherSuite1 (TLS Platform Profile 1) **device.sec.TLS.profile.cipherSuite2 (TLS Platform Profile 2)**	**string**	
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**	
The cipher suite to use for TLS Platform Profile 1 and TLS Platform profile 2. If set to 0, the custom cipher suite is used. If set to 1, the default cipher suite is used.		
device.sec.TLS.profile.deviceCert1 (TLS Platform Profile 1) **device.sec.TLS.profile.deviceCert2 (TLS Platform Profile 2)**	**Builtin, Platform1, Platform2**	
Choose the device certificate(s) for TLS Platform Profile 1 and TLS Platform Profile 2 to use for authentication.		
device.sec.TLS.profileSelection.dot1x	**PlatformProfile1, PlatformProfile2**	
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**	
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**	
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**	
The GMT offset—in seconds—to use for daylight savings time, corresponding to -12 to +13 hours.		
device.sntp.gmtOffsetcityID	**0 - 126**	**Null**
For descriptions of all values, refer to Set Time Zone Location Description.		
device.sntp.serverName	**IP address or domain name string**	
The SNTP server from which the phone obtains the current time.		

Device Parameters (continued)

Parameter	Permitted Values	Default
device.syslog.facility	**0 to 23**	
A description of what generated the log message. For more information, see RFC 3164.		
device.syslog.prependMac[1]	**0 or 1**	
If 1, the phone's MAC address is prepended to the log message sent to the syslog server.		
device.syslog.renderLevel[1]	**0 to 6**	
Specify the logging level that displays 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 to log. **0** or **1**: SeverityDebug(7). **2** or **3**: SeverityInformational(6). **4**: SeverityError(3). **5:** SeverityCritical(2). **6:** SeverityEmergency(0).		
device.syslog.serverName	**IP address or domain name string**	
The syslog server IP address or domain name string.		
device.syslog.transport	**None, UDP, TCP, TLS**	
The transport protocol that the phone uses to write to the syslog server. If set to None, transmission is turned off but the server address is preserved.		
device.wifi.country	**Two-letter country code**	**Null**
Enter the two-letter code for the country in which you are operating the RealPresence Trio 8800 solution with Wi-Fi enabled.		

[1] Change causes phone to restart or reboot.

Use these parameters to enable and set up the remote packet capture feature.

Remote Packet Capture Parameters

Parameter	Permitted Values	Default
diags.dumpcore.enabled[1]	**0 or 1**	**1**
When enabled, the phone generates a core file if it crashes. When disabled, the phone does not generate a core file when it crashes. The default value is 1, enabled.		
diags.pcap.enabled	**0 or 1**	**0**
Enable or disable all on-board packet capture features.		
diags.pcap.remote.enabled	**0 or 1**	**0**
Enable or disable the remote packet capture server.		

Remote Packet Capture Parameters (continued)

Parameter	Permitted Values	Default
diags.pcap.remote.password	**alphanumeric**	**<MAC Address>**
Enter the remote packet capture password.		
diags.pcap.remote.port	**Valid TCP Port**	**2002**
Specify the TLS profile to use for each application.		

[1] Change causes phone to restart or reboot.

The parameters listed in the following table 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.		

Dial Plan (Digit Map) Parameters (continued)

Parameter	Permitted Values	Default
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 turns 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 waits 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 is used. If there are more timeout values than digit maps, the extra timeout values are ignored.

Parameter	Permitted Values	Default
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.

Parameter	Permitted Values	Default
dialplan.impossibleMatchHandling[1]	**0, 1 or 2**	**0**

This parameter applies to digits you enter in dial mode which becomes active after you pick up the handset, headset, or press the New Call key.

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.

Parameter	Permitted Values	Default
dialplan.removeEndOfDial[1]	**0 or 1**	**1**

If set to 1, strip trailing # digit from digits sent out.

Parameter	Permitted Values	Default
dialplan.routing.emergency.outboundIdentity	**10-25 digits, a SIP, or a TEL URI**	**Null**

Choose how your phone is identified when you place an emergency call. You can use one of three formats: a 10-25 digit number, a valid SIP, or a TEL URI. If using a URI, the full URI is included verbatim in the P-A-I header. For example:

- dialplan.routing.emergency.outboundIdentity="5551238000"
- dialplan.routing.emergency.outboundIdentity=sip:john@emergency.com
- dialplan.routing.emergency.outboundIdentity="tel:+16045558000"

Dial Plan (Digit Map) Parameters (continued)

Parameter	Permitted Values	Default
dialplan.routing.emergency.preferredSource	**Config or ELIN**	**ELIN**

Use this parameter to set the precedence of the source of emergency outbound identities. When set to ELIN, the outbound identity used in the SIP P-Asserted-Identity header is taken from the network using an LLDP-MED Emergency Location Identifier Number (ELIN).
When set to Config, the parameter `dialplan.routing.emergency.outboundIdentity` has priority when enabled; the LLDP-MED ELIN value is used if `dialplan.routing.emergency.outboundIdentity` is null.

Parameter	Permitted Values	Default
dialplan.routing.emergency.x.description[1] **Emergency contact description**	**string**	**x=1:Emergency, Others: Null**
dialplan.routing.emergency.x.server.y[1] **Emergency server**	**positive integer**	**x=1: 1, others: Null**
dialplan.routing.emergency.x.value **Emergency URL values**	**SIP URL (single entry)**	**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 is 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.

Parameter	Permitted Values	Default
dialplan.routing.server.x.address[1]	**IP address or hostname**	**Null**

The IP address or hostname of a SIP server to use 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 to use for routing calls.

Parameter	Permitted Values	Default
dialplan.routing.server.x.transport[1]	**DNSnaptr, TCPpreferred, UDPOnly, TLS, TCPOnly**	**DNSnaptr**

The DNS lookup of the first server to be dialed is 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.

Parameter	Permitted Values	Default
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.

Dial Plan (Digit Map) Parameters (continued)

Parameter	Permitted Values	Default
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.

All of the parameters listed in the following table have a per-registration equivalent that you can configure. All of the per-registration parameters are listed in the following table. 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.

The maximum number of registrations (x) for each phone model is listed in the table Flexible Call Appearances under the column *Registrations*.

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
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
dialplan.x.originaldigitmap	string - max number of characters 2560	Null
dialplan.x.removeEndOfDial[1]	0 or 1	1

Per-Registration Dial Plan (Digit Map) Parameters (continued)

Parameter	Permitted Values	Default
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.

This parameter definition includes:

-   Polycom BroadSoft UC-One directory definitions
-   The corporate directory definition
-   Local contact directory for GENBAND
-   The local directory definition

Use the parameters listed in the following table with the Polycom BroadSoft UC-One directory.

Polycom BroadSoft UC-One Feature Parameters

Parameter	*Permitted Values*	*Default*
dir.broadsoft.regMap	**0 - Const_NumLineReg**	**1**
Specify the registration line credentials you want to use to retrieve directory information from BroadSoft UC-One directory when `dir.broadsoft.useXspCredentials=0`. This parameter is available with BroadSoft R20 Server or later.		
dir.broadsoft.useXspCredentials	**0 or 1**	**1**
If 1, the phone uses BroadSoft XSP credentials. If 0, the phone uses SIP credentials from `dir.broadsoft.regMap`.		
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`.		

Polycom BroadSoft UC-One Feature Parameters

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.		

Use the parameters in the following table 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]	**IP address, 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.alt.address	**hostname or FQDN**	**Null**
Enter the URL address of the GAB service provided by the server.		
dir.corp.alt.attribute.x.filter	**UTF-8 encoded string**	**Null**
Use a filter to set a predefined search string through configuration files.		
dir.corp.alt.attribute.x.label	**UTF-8 encoded string**	**Null**
Provide a label to identify a user.		
dir.corp.alt.attribute.x.name	**UTF-8 encoded string**	**Null**
The name of the parameter to match on the server. Each name must be unique; however, a global address book entry can have multiple parameters with the same name. You can configure up to eight parameters (x = 1 to 8).		
dir.corp.alt.attribute.x.sticky	**0 or 1**	**0**
Enable or disable a filter string. 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 '*' displays before the label of the attribute on the phone.		
dir.corp.alt.attribute.x.type	**first_name, last_name, phone_number, SIP_address, other**	**last_name**
Define how parameter x is interpreted by the phone. Entries can have multiple parameters of the same type. The value `other` is 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 global address book directory.		
dir.corp.alt.password	**UTF-8 encoded string**	**Null**
The password used to authenticate to the GENBAND server.		

Corporate Directory Parameters (continued)

Parameter	Permitted Values	Default
dir.corp.alt.port	**0, Null, 1 to 65535**	**0**
The port that connects to the server if a full URL is not provided.		
dir.corp.alt.protocol	**UTF-8 encoded string**	**sopi**
A directory protocol used to communicate to the corporate directory. The default value is sopi.		
dir.corp.alt.transport	**TCP or TLS**	**TCP**
A transport protocol used to communicate to the corporate directory. The default value is TCP.		
dir.corp.alt.user	**UTF-8 encoded string**	**Null**
The user name used to authenticate to the GENBAND server.		
dir.corp.alt.viewPersistence	**0 or 1**	**0**
Displays the results from your last address directory search.		
dir.corp.attribute.x.addstar[1]	**0 or 1**	**1**
If 1, the wildcard character, asterisk(*), is appended to the LDAP query field. If 0, the wildcard character, asterisk(*), is not appended to the query field.		
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 '*' displays 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.		

Corporate Directory Parameters (continued)

Parameter	Permitted Values	Default
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.		
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.customError	**UTF-8 encoded string**	**Null**
Configure the error message to display on the phone when the LDAP server finds an error.		
dir.corp.filterPrefix[1]	**UTF-8 encoded string**	**(objectclass=person)**
Predefined filter string for search queries.		
dir.corp.pageSize[1]	**8 to 64**	**64**
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.querySupportedControlOnInit	**0 or 1**	**1**
When enabled, the phone makes an initial query to check the status of the server when booting up.		
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**

Corporate Directory Parameters (continued)

Parameter	Permitted Values	Default
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.		
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.

The parameters listed in this section are for use with the GENBAND personnel address book (PAB) contact directory.

GENBAND Parameters

Parameter	Permitted Values	Default
dir.genband.local.contacts.maxSize	**1 - 100**	**100**
Specify the maximum number of contacts available in the GENBAND personnel address book contact directory.		

The next table lists parameters you can configure for your local contact directory. The maximum local directory size is limited based on the amount of flash memory in the phone and varies by phone model. The maximum number of contacts and maximum file size for phone models is listed in the table Maximum File Size and Number of Contacts. Polycom recommends that you configure a provisioning server that allows uploads to ensure a back-up copy of the directory when the phone reboots or loses power.

Note that on the VVX 1500, the local directory is by default stored in the phone's non-volatile device settings and you have to option to use the phone's volatile RAM and set the maximum file size.

Local Contact Directory Parameters

Parameter	Permitted Values	Default
dir.local.contacts.maxNum[1]	**VVX 300/301/310/311, 400/401/410/411, 500/501, 600/601 = 1-500** **VVX1500 = 1–9999** **SoundStructure VoIP Interface = not applicable**	**500** **9999**
Maximum number of contacts allowed in the local contact directory.		
dir.local.nonVolatile.maxSize	**1 - 100KB**	**VVX1500=100KB**
On the VVX 1500, set the maximum file size of the local contact directory stored on the phone's non-volatile memory.		
dir.local.passwordProtected	**0 or 1**	**0**
Specifies whether you are prompted for an Admin/User password when adding, editing, or deleting contacts in the Contact Directory. 0 (default) - No password prompt is displayed and pressing and holding the Line-key displays the Add/Edit menu. 1 - You are prompted for your Admin/User password while adding, editing, or deleting contacts in the 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. **Notes**: If provisioning polling is enabled and **dir.local.readonly=1** is enabled, the phone will look for a `<mac>-directory.xml` when matching the polling event. If provisioning polling is enabled and **dir.local.readonly=0** is disabled, the phone will not look for a `<mac>-directory.xml` when matching the polling event.		
dir.local.serverFeatureControl.method	**None or GENBANDSOPI**	**None**
Specify a method for synchronizing the directory and server. When set to GENBANDSOPI, the GENBANDSOPI protocol is enabled on the phone to get the personnel address book service from the GENBAND server.		
dir.local.serverFeatureControl.reg	**1 - 34**	**1**
Specify the phone line to enable personal address book feature on.		
dir.local.volatile	**0 or 1**	**0**
On the VVX 1500, enable or disable the use of volatile memory for the local contact directory. By default the VVX 1500 uses non-volatile memory.		
dir.local.volatile.maxSize	**1 - 200KB**	**VVX1500=200KB**
On the VVX 1500, set the maximum file size of the local contact directory stored on the phone's volatile memory.		
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.

The phone has a flexible call forward/diversion feature for each registration. In all cases, a call is diverted only if a non-Null contact has been configured.

The following numbers apply when no expansion module is attached. In the following table, x is the registration number. VVX 300 series: x=1-6; VVX 400 series: x=1-8; VVX 500series: x=1-12; VVX 600 series: x=1-16; VVX 1500: x=1-6. Attaching an expansion module increases the number to 34 (x=1-34).

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 are 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.autoOnSpecificCalle[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]	**0 or 1**	**1**
divert.busy.x.contact[1]	**contact address**	**Null**
Divert incoming calls that reach a busy signal. If `enabled` is set to 1, calls are diverted when registration x is busy. Calls are sent to the busy contact's address if it is specified; otherwise calls are sent to the default contact specified by `divert.x.contact`. If `enabled` is set to 0, calls are not diverted if the line is busy.		
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 are diverted when DND is enabled on registration x. Calls are sent to the DND contact's address if it is specified; otherwise calls are 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**

Call Diversion (Call Forwarding) Parameters (continued)

Parameter	*Permitted Values*	*Default*

If no-answer call diversion is `enabled`, calls that are not answered after the number of seconds specified by timeout are sent to the no-answer `contact`. If the no-answer `contact` is set to Null, the call is sent to the default contact specified by `divert.x.contact`. If `enabled` is set to 0, calls are 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.

The 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 following table. 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	**IP version 4 address**	**Null**
IP address.		
dns.cache.A.x.name	**valid hostname**	**Null**
Hostname		
dns.cache.A.x.ttl	**300 to 536870912 (2^29), seconds**	**300**

The TTL describes the time period the phone uses 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 retries 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 following table. 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 to look 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.

dns.cache.NAPTR.x.ttl	**300 to 536870912 (2^29), seconds**	**300**

The TTL describes the time period the phone uses 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 retries 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 following table. 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-SRV Parameters (continued)

Parameter	Permitted values	Default
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 uses 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 retries 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.		

Use the following tables to configure the enhanced feature key (EFK) feature on your phone:

- Enhanced Feature Key (EFK) Version Parameters
- Enhanced Feature Key (EFK) List Parameters
- Enhanced Feature Key (EFK) Prompt Parameters
- Enhanced Feature Key (EFK) Soft Key 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.		

In the following table, the registration line x=1-50.

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 performs. 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 the section Understand Macro Definitions.		
efk.efklist.x.label	**string**	**Null**
The text string 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 is shortened and '…' is 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.		

In the following table, the registration line x=1-50.

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 is shortened and '…' is appended.		
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 fails.		
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.		

Enhanced Feature Key (EFK) Prompt Parameters (continued)

Parameter Name	Permitted Values	Default
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.

Enhanced Feature Key (EFK) Soft Key Parameters

Parameter	Permitted Values	Default
efk.softkey.alignleft	0 or 1	0
Use this parameter to left-align soft keys and remove blank soft keys from the order. By default, this parameter is disabled. Setting this parameter to 1 left-aligns soft keys and removes blank soft keys from the order. Note: This parameter does not work with custom soft keys.		

The exchange parameter controls aspects of meeting invitations.

Exchange Parameters

Parameter	Permitted Values	Default
exchange.meeting.parseOption[1]	All, Location, LocationAndSubject, Description, Enum	Location
Indicates the field in the meeting invite from which the VMR or meeting number should be fetched.		

[1] Change causes phone to restart or reboot.

The feature parameters listed in the following table 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 Activation/Deactivation Parameters (continued)

Parameter	Permitted Values	Default
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.advancedConference.enabled	**0 or 1**	**0**

Enable or disable advanced conferences and conference controls for ALU advanced conferences.
If enabled, conference controls display. If disabled, conference controls do not display.

feature.audioVideoToggle.enabled	**0 or 1**	**0**

Applies to the VVX 1500 and VVX camera-enabled VVX 500/501 and 600/601 business media phones. If 0, the audio/video toggle feature is disabled. If 1, the feature is enabled.

feature.bluetooth.enabled	**0 or 1**	**1**

VVX 600/601 high-security environments. If 0, the Bluetooth feature is disabled. If 1, Bluetooth is enabled.

feature.broadsoftdir.enabled[1]	**0 or 1**	**0**

If 1, the BroadSoft enterprise directory is enabled. If 0, the directory is disabled

feature.broadsoftUcOne.enabled[1]	**0 or 1**	**0**

If 1, the BroadSoft UC-One feature is enabled. If 0, the feature is disabled.

feature.broadsoft.xsi.AnonymousCallReject.enabled	**0 or 1**	**0**

Displays the Anonymous Call Rejection menu on the phone. If set to 1, the Anonymous Call Rejection menu displays and the user can turn the feature on or off from the phone. If set to 0, the Anonymous Call Rejection menu does not display to users

feature.broadsoft.xsi.BroadWorksAnywhere.enabled	**0 or 1**	**0**

Enable or disable the BroadWorks Anywhere feature menu on the phone. If set to 0, the feature menu is disabled does not display.

feature.broadsoft.xsi.LineIdblock.enabled	**0 or 1**	**0**

Enable or disable the Line ID Blocking feature menu on the phone. If set to 0, the feature menu is disabled and does not display on the phone.

feature.broadsoft.xsi.RemoteOffice.enabled	**0 or 1**	**0**

Enable or disable the Remote Office feature menu on the phone. If set to 1, the feature menu is enabled and displays on the phone.

Feature Activation/Deactivation Parameters (continued)

Parameter	Permitted Values	Default
feature.broadsoft.xsi.SimultaneousRing.enabled	**0 or 1**	**0**
Enable or disable the Simultaneous Ring Personal feature menu on the phone. If set to 0, the feature menu is disabled and does not display.		
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.		
feature.callRecording.enabled[1]	**0 or 1**	**0**
Available for devices with a USB port. If 0, the call recording and playback feature is disabled. If 1, the feature is enabled.		
feature.contacts.enabled	**0 or 1**	**1**
Enable or disable display of the Contacts icon displays on the Home screen, the global menu, and in the dialer. Requires UCS 5.4.2 RevAA or higher.		
feature.corporateDirectory.alt.enabled	**0 or 1**	**0**
Enable or disable the global address book service.		
feature.corporateDirectory.enabled	**0 or 1**	**0**
If 0, the corporate directory feature is disabled. If 1, the feature is enabled.		
feature.dect.enabled	**0 or 1**	**0**
Enables or disables communication and pairing with the VVX D60 Wireless Handset and Base Station accessories. When enabled, the VVX D60 menu options display on the phone and in the Web Configuration Utility. When disabled, the VVX D60 menu options do not display.		
feature.directedCallPickup.enabled[1]	**0 or 1**	**0**
If 0, the directed call pickup feature is disabled. If 1, the feature is enabled.		

Feature Activation/Deactivation Parameters (continued)

Parameter	Permitted Values	Default
feature.directory.enabled	**0 or 1**	**1**
If 0, the local contact directory is disabled. If 1, the directory is enabled.		
feature.doNotDisturb.enable[1]	**0 or 1**	**1**
Enable or disable do not disturb (DND). When disabled, the DND soft key does not display and the option is removed from the phone's menu system at Menu > Settings > Features.		
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 displays the number only (for example, 1234567).		
feature.enhancedCallPark.allowAudioNotification	**0 or 1**	**0**
Enables and disables the audio notifications for parked calls on private and shared lines.		
feature.enhancedFeatureKeys.enabled	**0 or 1**	**0**
If 0, the enhanced feature keys feature is disabled. If 1, the feature is enabled.		
feature.exchange2007.interop.enabled	**0 or 1**	**0**
If 0, interoperability with Exchange 2007 is disabled. If 1, interoperability with Exchange 2007 is enabled. Note: If enabled, `feature.exchangeVoiceMail.enabled` should be disabled.		
feature.exchangeCalendar.enabled[1]	**0 or 1**	**1**
For the VVX 500/501, 600/601 and 1500 phones. If 1, the Exchange Calendar feature is enabled as a service. If 0, the feature is disabled.		
feature.flexibleLineKey.enable	**0 or 1**	**Lync = 0**
Enables and disables the Lync Flexible Line Key feature. Not available on the VVX 101, 201, or 1500 business media phones.		
feature.forward.enable	**0 or 1**	**1**
Enable or disable call forwarding. When disabled, the Forward soft key does not display and the option is removed from the phone's menu system at Menu > Settings > Features.		
feature.genband.E911.enabled[1]	**0 or 1**	**0**
Enable or disable the GENBAND E.911 feature.		
feature.groupCallPickup.enabled[1]	**0 or 1**	**0**
If 0, the group call pickup feature is disabled. If 1, the SIP-B group call pickup feature is enabled.		
feature.hoteling.enabled	**0 or 1**	**0**
If 0, Hoteling is disabled. If 1, Hoteling is enabled.		
feature.intercom.enable	**0 or 1**	**0**

Feature Activation/Deactivation Parameters (continued)

Parameter	Permitted Values	Default
Enable or disable the intercom feature.		
feature.lastCallReturn.enabled[1]	**0 or 1**	**0**
If 0, the last call return feature is disabled. If 1, the feature is enabled.		
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.		
feature.nWayConference.enabled	**0 or 1**	**0**
If 0, the n-way conferencing managing feature is disabled; you can hold three-way conferences but the options to manage the conference do not display. If 1, n-way conferencing is enabled, you can hold conferences with the maximum number of parties, and the options to manage the conference display.		
feature.persistentMute.enabled[1]	**0 or 1**	**0**
Set to 1 to enable the persistent mute feature. When set to 0, mute ends when the active call ends or when the phone restarts.		
feature.pictureFrame.enabled[1]	**0 or 1**	**1**
For the VVX 500/501, 600/601, and 1500 only. If 0, the digital picture frame feature is disabled. If 1, the digital picture frame feature is enabled.		
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. Note that the UC-One directory user interface uses QML as the UI framework and the viewer is used to load the QML applications.		
feature.ringDownload.enabled[1]	**0 or 1**	**1**
If 0, the phone does not download ringtones when it starts up. If 1, the phone downloads ringtones when it starts up.		
feature.scap.defCallTypeExclusive	**0 or 1s**	**1s**
Controls the default behavior of a Shared Call Appearance call. By default, an outgoing call from the call group is private. After the call is answered, the user needs to press Share soft key to make the call public so that other people on the line can bridge in to the call.		
feature.scap.HoldRequestUriUserPart	**string**	**SCAP-Hold**
Specifies the Hold request for Shared Call Appearance calls to the ALU server. This value must match the value configured on ALU server for SCA hold request.		

Feature Activation/Deactivation Parameters (continued)

Parameter	Permitted Values	Default
feature.uniqueCallLabeling.enabled[1]	**0 or 1**	**0**
If 0, disables the unique call labeling feature. If 1, enables the unique call labeling feature. Use reg.x.line.y.label to define unique labels.		
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 are identified on the display by their phone's IP address.		
feature.usbRear.power.enabled	**0 or 1**	**1**
If 1, power to the rear USB port (port 2) is enabled. If 0, power to the rear USB port is disabled and the phone does not detect USB devices to the rear USB port. Note: This parameter does not apply to VVX 1500 phones.		
feature.usbTop.power.enabled	**0 or 1**	**1**
If 1, power to the top USB port (port 1) is enabled. If 0, power to the top USB port is disabled and the phone does not detect USB devices to the top USB port.		
feature.VDP.enabled[1]	**0 or 1**	**0**
If 1, VDP is enabled and the phone displays the Visitor Login soft key. If 0, VDP is disabled and the phone does not display the Visitor Login soft key.		
feature.VVXD60.allowLineMapppings	**0 or 1**	**0**

---

[1] Change causes phone to restart or reboot.

The following table lists GENBAND E.911 parameters.

GENBAND Parameters

Parameter	Permitted Values	Default
genband.E911.location.description	**String up to 256 characters [platform-specific display size limitations apply]**	**Other**
Enter a description of the location of the phone, for example, cubicle 105. Note: Ensure that the description string you provide here is identical to the description you configure on the location server.		
genband.E911.location.locationID	**1 to 256**	**0**
Enter the location ID corresponding to the location description you entered in genband.E911.location.description, for example, 112876. Note: Ensure that the location ID you enter here is identical to the one you configure on the location server.		

GENBAND Parameters (continued)

Parameter	Permitted Values	Default
genband.E911.registration.line	**0 to 100**	**1**
Select the registration line to use to retrieve E.911 location information		

The parameters in this section control the display of icons on the phone's Home screen.

Homescreen Parameters

Parameter	Permitted Values	Default
homescreen.intercom.enable	**0 or 1**	**1**
Enable or disable the Intercom icon on the device home screen.		
homescreen.status.enable	**0 or 1**	**1**
Enable or disable the display of the Status menu icon on the Home screen.		

The hoteling enhancement to ACD enables agents to use any available host phone by logging in with agent credentials. After logging in, agents have access to their guest profile and ACD settings on the host phone. The hoteling enhancement is independent of the ACD feature, meaning agents can use hoteling whether the premium ACD feature is enabled or disabled. The following table lists parameters you can configure.

Hoteling Parameters

Parameter	Permitted Values	Default
hoteling.reg	**1 – 34**	**1**
If ACD is disabled, the phone uses this line registration index for hoteling. If ACD is enabled, this value must be the same as the ACD line registration index.		

The phone contains a local Web Configuration Utility server for user and administrator features. Note that several of these parameters can be used with Microsoft Skype for Business Server and the parameter values listed in the table Enable Web Configuration Utility have two default states: a generic default value for UC Software 5.1.0 and a different value when the phone is registered with Skype for Business Server. The following table lists the default values for both states where applicable.

The web server supports both basic and digest authentication. The authentication user name and password are not configurable for this release.

HTTPD (Web Server) Parameters

Parameter	Permitted Values	Default
httpd.enabled[1]	**0 or 1**	**Generic=1** **Lync=0**
If 0, the HTTP server is disabled (the Web Configuration Utility is also be disabled). If 1, the server is enabled. Note: This parameter must be enabled to take screen captures of the phone screen.		
httpd.cfg.enabled[1]	**0 or 1**	**Generic=1** **Lync=0**
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.		
httpd.cfg.secureTunnelEnabled[1]	**0 or 1**	**Generic=1** **Lync=0**
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**	**1**
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.

The following table lists parameters that configure the phone's Home screen display.

Homescreen Parameters

Parameter	Permitted Values	Default
homeScreen.application.enable	**0 or 1**	**1**
Enable or disable display of the Applications icon on the phone Home screen.		
homeScreen.calendarf.enable	**0 or 1**	**1**
Enable or disable display of the Calendar icon on the phone Home screen.		
homeScreen.directories.enable	**0 or 1**	**1**
Enable or disable display of the Directories menu icon on the phone Home screen.		
homeScreen.doNotDisturb.enable	**0 or 1**	**1**

Homescreen Parameters (continued)

Enable or disable display of the DND icon on the phone's Home screen.

| homeScreen.features.enable | 0 or 1 | 1 |

Enable or disable display of the Features menu icon on the phone Home screen.

| homeScreen.forward.enable | 0 or 1 | 1 |

Enable or disable display of the call forward icon on the phone Home screen.

| homeScreen.messages.enable | 0 or 1 | 1 |

Enable or disable display of the Messages menu icon on the phone Home screen.

| homeScreen.newCalls.enable | 0 or 1 | 1 |

Enable or disable display of the New Call icon on the phone Home screen.

| homeScreen.settings.enable | 0 or 1 | 1 |

Enable or disable display of the Settings menu icon on the phone Home screen.

| homeScreen.UCOne.enable | 0 or 1 | 0 |

Enable or disable the UC-One Settings icon to display on the Home screen.

This section explains how to configure the LED indicators on the RealPresence Trio system, VVX business media phones and expansion modules.

The table Indicator Pattern Parameters lists parameters that enable you to set the patterns, color, and duration of the LED indicators on VVX phones and expansion modules.

The table LED Indicator Pattern Type lists the values that configure the LED indicator pattern type where *x* is the pattern type and *y* is the pattern number. For y, enter a value of 1-20 to indicate the pattern number.

Indicator Pattern Parameters

Parameter	*Permitted Values*	*Default*
ind.pattern.x.step.y.state	**On or Off**	**On**

Turns the LED indicator on or off depending on the pattern's state.

| **ind.pattern.x.step.y.color** | **Red, Green, or Yellow** | **Red** |

Indicates the color of the line key LED indicators. The Yellow value is only available for VVX 300 and 400 series phones. You cannot set the line key LED indicator to Yellow on VVX 101 and 201 phones or on the VVX Expansion Modules.

Note: You can only configure the color of the LED indicators for line keys only. You cannot configure the color of the LED indicators for the Headset key or MWI.

| **ind.pattern.x.step.y.duration** | **Positive integer** | **0 = infinite** |

Enter the duration of the pattern of the LED indicator.

LED Indicator Pattern Type

Pattern Type	Function
powerSaving	Sets the behavior for Message Waiting Indicator when the phone is in Power Saving mode.
active	Sets the pattern for line keys during active calls.
on	Turns on the LED indicator pattern.
off	Turns off the LED indicator pattern.
offering	Sets the pattern for line keys during incoming calls.
flash	Sets the pattern for line keys during held calls and the Message Waiting Indicator when there are unread voicemail messages.
lockedOut	Sets the pattern for line keys when a remote party is busy on a shared line.
FlashSlow	Sets the pattern for the Headset key when Headset Memory Mode is enabled.
held	Sets the pattern for line keys during a held call.
remoteBusyOffering	Sets the pattern for line keys for monitored BLF contacts.

<key/>

The next table lists parameters that enable you to change the default functions of your phone's keypad keys, a process also known as remapping. If you want to change the default function of a key, you must specify the phone model number, the key you want to change, and a new function for the key.

● For a list of products and their model codes, see the section Product, Model Code, and Part Number Mapping.

● To find the key number, location of the key on each phone model, and default key functions, refer to the section Define the Phone Key Layout.

● For a list of parameter values you can assign as functions to a phone key, refer to the table Keypad Key Functions.

> **Caution: Key remapping is not recommended**
> Polycom does not recommend remapping or changing the default functions of the keys on your phone.

Key Parameters

Parameter	Permitted Values
key.x.y.function.prim	For a list of functions refer to the table Keypad Key Functions

Specify a phone model, key number, and function.

x can be one of the VVX 300 series, 400 series, 500 series, 600 series, or VVX1500 phones.

y can be one key number.

[1] Change causes phone to restart or reboot.

The following table lists the functions that are available for phone keys.

Keypad Key Functions

Answer	Dialpad2	Handsfree	MyStatus	SpeedDialMenu
ArrowDown	Dialpad3	Headset	Null	Talk
ArrowLeft	Dialpad4	Hold	Offline	Video
ArrowRight	Dialpad5	Home	Redial	VolDown
ArrowUp	Dialpad6	Line2	Release	VolUp
Back	Dialpad8	Line3	Select	
BuddyStatus	Dialpad9	Line4	Setup	
CallList	DialpadStar	Line5	SoftKey1	
Conference	DialPound	Line6	SoftKey2	
Delete	Directories	Messages	SoftKey3	
Dialpad0	DoNotDisturb	Menu	SoftKey4	
Dialpad1	Green	MicMute	SpeedDial	

Example Custom Key Configurations

This section provides several custom key configuration examples.

To remap the volume up key to answer a call on the VVX 300:

» Update the configuration file as follows: `key.VVX300.6.function.prim="Answer"`

» **To remap the volume down key to launch the Settings menu on the VVX 300 using a macro:**

» Update the configuration file as follows:

➢ `key.VVX300.7.function.prim="$Msetting$"`

➢ `efk.efklist.1.action.string="$FSetup$"`

> ➤ `efk.efklist.1.mname="setting"`
> ➤ `efk.efklist.1.status="1"`

To remap the Mute key to launch the Forward Menu on the VVX 500 using EFK.

» Update the configuration file as follows: `key.VVX500.18.function.prim="$FDivert$"`

To remap the Transfer key to phone lock using an EFK macro:

» Update the configuration file as follows: `key.37.function.prim="$FLockPhone$"`

To remap the Redial key:

» Update the configuration file as follows:

```
key.36.function.prim="http://vanoem02.vancouver.polycom.com:8080/MicroBrowserTest.html"
```

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 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:

● The multilingual definitions
● The date and time definitions

The multilingual parameters listed in the following table are 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) is used, otherwise, the language to be used may be specified in the format of `lcl.ml.lang.menu.x.label`. For example, to boot up the phone in German, set this parameter to `German_Germany`.

lcl.ml.lang.charset[1]	**string**

The language character set.

lcl.ml.lang.clock.x.24HourClock	**0 or 1**

This parameter 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**

This parameter 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.japanese.font.enable d	**0 or 1**

This parameter applies to VVX 400, 401, 410, 411, 500, 501, 600, 601, and 1500.

0 (default) - The phone does not use Japanese Kanji character font.

1 - The phone displays Japanese Kanji character font.

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**	**String in the format language_region**
lcl.ml.lang.menu.x.label[1] **Phone language menu label**	**String in the format nativeLanguageName (abbreviation)**

Multilingual Parameters (continued)

Parameter	Permitted Values
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 **VVXLocalization** 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 is: `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

4 `language_region` parallel to the other dictionary files under the VVXLocalization folder on the provisioning server.

5 Add an `lcl.ml.lang.clock.menu.x` parameter to the configuration file.

6 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.

7 (Optional) Set `lcl.ml.lang` to be the new `language_region` string.

The basic character support includes the Unicode character ranges listed in the next table.

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

The parameters listed in the following table 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. If 0, display time above date.

Date and Time Parameters (continued)

Parameter	Permitted Values	Default
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.

Parameter	Permitted Values	Default
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).

Parameter	Permitted Values	Default
lcl.datetime.time.24HourClock	**0 or 1**	

If set to 1, display time in 24-hour clock mode rather than a.m./p.m.

The parameters listed in the next table enable you to configure the feature licensing system.

Feature License Parameters

Parameter	Permitted Values	Default
license.polling.time[1]	**00:00 – 23:59**	**02:00**

The time (using the 24-hour clock) to check if the license has expired.

[1] Change causes phone to restart or reboot.

> **Note: Removing the installed license**
> Once the license is installed on a phone, it cannot be removed.

The parameters listed in the next table are available for the Flexible Line Key Assignment.

Line Key Parameters

Parameter	Permitted Values	Default
lineKey.x.category[1]	**unassigned, line, BLF, speedDial, presence**	**unassigned**

The line key category. Set the category to unassigned to leave a blank line key.

Line Key Parameters (continued)

Parameter	Permitted Values	Default
lineKey.x.index[1]	**0 to 9999**	**0**
For lines, the index for line numbers. For speed dials, the speed dial index. For BLF or presence, 0. For unassigned, the value is ignored.		
lineKey.reassignment.enabled[1]	**0 or 1**	**0**
If 1, flexible line key assignment is enabled.		

[1] Change causes phone to restart or reboot.

The parameters listed in the following table enable you to configure settings for LLDP discovery.

LLDP Parameters

Parameter	Permitted Values	Default
lldpFastStartCount	**3 to 10**	**5**
Specifies the number of consecutive LLDP packets the phone sends at the time of LLDP discovery. Note that LLDP packets are sent every one second.		

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
-

> **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

Logging Levels (continued)

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 next table.

Event Timestamp Formats

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 following table.

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, bsdir, cap, cdp, cert, cfg, cipher, clink, clist, cmp, cmr, copy, curl, daa, dapi, dbs, dbuf, dhcpc, dis, dock, dot1x, dns, drvbt, ec, efk, ethf, flk, h323, hset, httpa, httpd, hw, ht, ib, key, ldap, lic, lldp, loc, log, mb, mobil, net, niche, ocsp, osd, pcap, pcd, pdc, peer, pgui, pmt, poll, pps, pres, pstn, ptt, push, pwrsv, rdisk, res, rtos, rtls, sec, sig, sip, slog, so, srtp, sshc, ssps, style, sync, sys, ta, task, tls, trace, ttrs, usb, usbio, util, utilm, vsr, wdog, wmgr, and xmpp.		
log.level.change.dect	**0-6**	**4**
Sets the logging detail level for the VVX D60 accessory.		
log.level.change.flk	**0 - 6**	**4**
Sets the log level for Lync FLK logs.		
log.level.change.mr	**0 - 6**	**4**
Initial logging level for the Networked Devices log module.		
log.level.change.mraud	**0 - 6**	**4**

Logging Level, Change, and Render Parameters (continued)

Parameter	Permitted Values	Default
Initial logging level for the Networked Devices Audio log module.		
log.level.change.mrmgr	**0 - 6**	**4**
Initial logging level for the Networked Devices Manager log module.		
log.level.change.prox	**0 - 6**	**4**
Initial logging level for the Proximity log module.		
log.level.change.ptp	**0 - 6**	**4**
Initial logging level for the Precision Time Protocol log module.		
log.level.change.sopi	**0 - 6**	**4**
Specify the SOPI service log level for the GENBAND Global Address Book and Personnel Address Book.		
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 - 10240**	**512**
Maximum size of flash memory for logs in Kbytes. When this size is about to be exceeded, the phone uploads all logs that have not yet been uploaded, and erases half of the logs on the phone. The administrator can use a web browser to read 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.		
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 is not uploaded if no new events have been logged since the last upload.		

Logging Level, Change, and Render Parameters (continued)

Parameter	Permitted Values	Default
log.render.level	**0 to 6**	**1**

Specifies the lowest class of event 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)

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**	**0**

Set to 0. Polycom recommends that you do not change this value.

log.render.type	**0 to 2**	**2**

Refer to the table Event Timestamp Formats for timestamp type.

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 next table 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	Default Value
log.sched.x.level	**0 to 5**	**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**	**15**

Seconds between each command execution. 0=run once

log.sched.x.startDay	**0 to 7**	**7**

When startMode is abs, specifies the day of the week to start command execution. 1=Sun, 2=Mon, ..., 7=Sat

Logging Schedule Parameters (continued)

Parameter	Permitted Values	Default Value
log.sched.x.startMode	**0 - 64**	
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.		

The next table lists parameters that configure the home page, proxy, and size limits used by the microbrowser and browser when selected to provide services. The microbrowser and web browser are supported on the VVX 300 series, 400 series, 500 series, 600 series, and 1500 phones.

Microbrowser and Web Browser Parameters

Parameter	Permitted Values	Default
mb.idleDisplay.home	**Null or any fully formed valid HTTP URL. Length up to 255 characters.**	**Null**
The URL for the microbrowser home page that is shown on the idle display microbrowser Home page. For example: http://www.example.com/xhtml/frontpage. If Null, there is no idle display microbrowser. Note that the microbrowser idle display displaces the idle display indicator.		
mb.idleDisplay.refresh	**0 or an integer > 5**	**0**
The time period in seconds that the microbrowser's idle display refreshes. If set to 0, the idle display microbrowser does not refresh. The minimum refresh period is 5 seconds (values from 1 to 4 are ignored, and 5 is used). Note: If an HTTP Refresh header is detected, it is respected, even if this parameter is set to 0. The refresh parameter is respected only in the event that a refresh fails. Once a refresh is successful, the value in the HTTP refresh header, if available, is used.		
mb.idleRefresh.onFailure	**60 - 655350 seconds**	**60 seconds**
To reduce requests from the phone when the idle display server is unavailable, specify a delay in seconds the phone sends refresh requests to the idle browser server when unavailable. This delay applies only when the server returns HTTP 5xx errors. To control the refresh times when the server is functioning, use `mb.idleDisplay.refresh`.		
mb.main.autoBackKey[1]	**0 or 1**	**1**
If 0, the phone does not provide a **Back** soft key; all soft keys are created and controlled by the application. If 1, the phone automatically supplies a **Back** soft key in all main browser screens. The **Back** soft key takes the user back to the previous page in the browser history.		
mb.main.home	**Any fully formed valid HTTP URL. Length up to 255 characters.**	**Null**
The URL of the microbrowser's home page. For example: `http://www.example.com/xhtml/frontpage/home`. If blank, the browser notifies the user that a blank home-page was used.		

Microbrowser and Web Browser Parameters (continued)

Parameter	Permitted Values	Default
mb.main.idleTimeout	**0 - 600, seconds**	**40**
The timeout, in seconds, for the interactive browser. If the interactive browser remains idle for the defined period of time, the phone returns to the idle browser. If 0, there is no timeout.		
mb.main.loadWebImages	**0 or 1**	**1**
If 0, disables loading of images in a browser. If 1, images can load in a browser.		
mb.main.reloadPage	**0 or 1**	**0**
If 0, the microbrowser displays the content of the most recently viewed web page. If 1, the microbrowser loads the URL configured in **mb.main.home** each time the browser is launched.		
mb.main.statusbar	**0 or 1**	**0**
If 0, the status bar does not display. If 1, the status bar displays and status messages are shown.		
mb.main.toolbar.autoHide.enabled	**0 or 1**	**1**
If 0, the toolbar displays continually. If 1, the toolbar disappears if not selected.		
mb.proxy	**Null or domain name or IP address in the format <address>:<port>**	**Null. Default port = 8080**
The address of the HTTP proxy to be used by the microbrowser. If blank, normal unproxied HTTP is used by the microbrowser.		
mb.ssawc.call.mode	**active or passive**	**passive**
Control the spontaneous display of web content. If set to `passive`, web content is displayed only when requested by the user. If set to `active`, web content is displayed immediately.		
mb.ssawc.enabled	**0 or 1**	**0**
If 0, spontaneous display of web content is disabled. If 1, spontaneous web content display is enabled.		

[1] Change causes phone to restart or reboot.

The next table lists parameters you can use to configure the message-waiting feature, which is supported on a per-registration basis.

The maximum number of registrations (x) for each phone model is listed in the table Flexible Call Appearances under the column *Registrations*.

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** or **MSG** key. If 0, the phone shows the menus Message Center and Instant Messages. If 1, the phone bypasses these menus and goes directly to voicemail. This parameter applies only to phone models that have a Messages or MSG key.		
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 sends a SUBSCRIBE request to this contact after boot-up.		
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 does **not** flash when there are new unread messages for the selected line. When set to 1, the LED flashes 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.

The parameters listed in the next table 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 enabled, the backlight is not illuminated on new voice message arrival. By default this parameter is disabled.		

The parameters listed in the next table define port and IP address changes used in NAT traversal. The port changes alter 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]	**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 sends a keepalive 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 does not send out keepalive messages.		
nat.mediaPortStart[1]	**0 to 65440**	**0**
The initially allocated RTP port. Overrides the value set for `tcpIpApp.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.

The parameters listed in this section control the Ethernet interface maximum transmission unit (MTU) on VVX business media phones.

Ethernet Interface MTU Parameters

Parameter	Permitted Values	Default
net.interface.mtu	**800 - 1500**	**1496 (for RealPresence Trio 8800 and RealPresence Trio Visual+)**
Configure the Ethernet or Wi-Fi interface maximum transmission unit (MTU) on VVX business media phones or RealPresence Trio solution. Note that this parameter affects the LAN port and the PC port.		
net.lldp.extendedDiscovery	**0 to 3600**	**0**
Specify the duration of time that LLDP discovery continues after sending the number of packets defined by the parameter `lldpFastStartCount`. Note that LLDP packets are sent every 5 seconds during this extended discovery period.		

The parameters listed in the next table must be enabled if you want to use the Lock soft key.

Phone Lock Parameters

Parameter	Permitted Values	Default
phoneLock.Allow.AnswerOnLock	**0 or 1**	**1**
If 1, the phone answers any incoming call without asking to UNLOCK. If 0, the phone asks to UNLOCK before answering.		
phoneLock.authorized.x.description **The name or description of an authorized number** **phoneLock.authorized.x.value** **The number or address for an authorized contact**	**String** **string**	
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.		
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.

The power-saving feature automatically turns off the phone's LCD display when not in use. This feature is disabled by default on the VVX 300 and 400 series phones, and enabled by default on the VVX 500 series, 600 series, and 1500 phones. The parameters `powerSaving.userDetectionSensitivity.*` listed in the next table are supported only on the VVX 1500 business media phones.

Power Saving Parameters

Parameter	Permitted Values	Default
powerSaving.enable	**0 or 1**	**VVX 201=0** **VVX 300/301/310/311=0** **VVX 400/401/410/411=0** **VVX 500/501, 600/601, 1500=1**
If 0, the LCD power saving feature is disabled. If 1, the feature is enabled. The power-saving feature is disabled by default on the VVX 201 and VVX 300- and 400-series phones, and is enabled by default on the VVX 500 series, 600 series, and 1500. Note that when the phone is in power-saving mode, the LED Message Waiting Indicator (MWI) flashes. To disable the MWI LED when the phone is in power saving mode, set the parameter `ind.pattern.powerSaving.step.1.state.x` to 0 where x=your phone's model. For example, enter the parameter as `ind.pattern.powerSaving.step.1.state.VVX500` to disable the MWI for your VVX 500 phone.		
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, VVX 1500=10**
The number of minutes to wait while the phone is idle during office hours before activating power saving. Note that the default time for VVX 300 series, 400 series, 500 series, and 600 series is 480 minutes. The default time for the VVX 1500 is 10 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.		
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).		

Power Saving Parameters (continued)

Parameter	Permitted Values	Default
powerSaving.userDetectionSensitivity.offHours	**0 to 10**	**2**
Available on the VVX 1500 only. The sensitivity 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**
Available on the VVX 1500 only. The sensitivity 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.		

The next table 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 uses 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.		
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.		

The parameters listed in the next table control the provisioning server system for your phones.

Provisioning Parameters

Parameter	Permitted Values	Default
prov.autoConfigUpload.enabled	**0 or 1**	**1**

Enable or disable the automatic upload of phone and Web Configuration Utility override configuration files to the provisioning server. By default, per-phone `MAC-phone.cfg` and `MAC-web.cfg` files are automatically uploaded to the provisioning server when a configuration change is made from the phone interface or Web Configuration Utility respectively. When disabled, per-phone override files are not uploaded to the provisioning server.

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.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.

prov.login.password.encodingMode	**Abc, ABC, abc, 123**	**123**

This parameter configures the default encoding mode for the text in the Password field in User Login screen.

prov.login.userId.encodingMode	**Abc, ABC, abc, 123**	**Abc**

Configures the default encoding mode for the text in the User ID field in User Login screen.

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 does 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**

Provisioning Parameters (continued)

Parameter	Permitted Values	Default
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.)		
prov.quickSetup.limitServerDetails	**0 or 1**	**0**
If 1, a screen to enter only user name and password is shown. Other details are taken from `ztp/dhcp` (option66). If 0, user must provide all the details, for example, DHCP option, server address, server type) in addition to user name and password.		
prov.usercontrol.enabled	**0 or 1**	**1**
When this parameter is set to 1, the phone displays the software update notification and options, and the user can control the software download. If set to 0, the phone does not displays the software update notification and options, and the phone reboots automatically to update the software.		
prov.usercontrol.postponeTime	**15 minutes, 1hour, 2 hours, 4 hours, 6 hours**	**2 hours**
Configure a time interval for software update notications. Permitted values for this configuration parameter are 15 min, 1 hour, 2hours, 4 hours and 6 hours using the format HH:MM. If a user configures an invalid value the default value is used.		
prov.vdp.accessCode.login	**string**	***771**
Specify the VDP login service access code.		
prov.vdp.accessCode.logout	**string**	***772**
Specify the VDP logout service access code.		

[1] Change causes phone to restart or reboot.

The PTT (push-to-talk) parameter is used to configure Push-to-Talk features. The parameters in the next table configure the PTT mode and page mode features.

Push-To-Talk and 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.callWaiting.enable	**0 or 1**	**0**
If 0, incoming PTT sessions do not produce standard call waiting. If 1, incoming PTT sessions produce standard call waiting behavior on the active audio channel.		
ptt.compatibilityMode	**0 or 1**	**0**
If 0, the PTT/paging codec used is controlled by the ptt.codec and ptt.pageMode.codec parameters, allowing codecs to be chosen that are not compatible with SpectraLink handset models 8020/8030 or older.		
If 1, the PTT/paging codec used will always be compatible with SpectraLink handset models 8020/8030 or older, even if other configuration parameters are incompatible. For example, if this parameter is enabled and ptt.codec is set to G.722, the G.726QI codec is used for outgoing PTT audio to maintain compatibility.		
ptt.emergencyChannel.volume	**-57 to 0**	**-10**
The volume of emergency pages relative to the maximum speakerphone volume of the phone. Positive values are louder than the maximum and negative values are quieter. The gain to use for emergency page/PTT is the maximum termination gain plus this parameter. **Note**: To enter a negative number, press the * key first.		
ptt.port	**0 to 65535**	**5001**
The port to send audio to and receive audio from.		
ptt.pageMode.allowOffHookPages	**0 or 1**	**0**
If 0, group pages do not play out on the phone during an active call—except for Priority and Emergency pages. If 1, group pages play out on the handset during an active call.		
ptt.pageMode.codec	**G.711Mu, G.726QI, or G.722**	**G.722**
The audio codec to use for outgoing group pages. Incoming pages are decoded according to the codec specified in the incoming message.		
ptt.pageMode.defaultGroup	**1 to 25**	**1**
The paging group used to transmit an outgoing page if the user does not explicitly specify a group.		
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` is used.		
ptt.pageMode.emergencyGroup	**1 to 25**	**25**
The paging group to use for emergency pages.		

Push-To-Talk and Group Paging Parameters (continued)

Parameter	Permitted Values	Default
ptt.pageMode.enable	**0 or 1**	**0**
If 0, group paging is disabled. If 1, group paging is enabled.		
ptt.pageMode.group.x.available	**0 or 1**	**1**
Make the group available to the user.		
ptt.pageMode.group.x.allowReceive	**0 or 1**	**1**
If 0, phone cannot receive pages on the specified group. If 1, phone can receive pages on the specified group.		
ptt.pageMode.group.x.allowTransmit	**0 or 1**	**1**
Allow outgoing announcements to the group		
ptt.pageMode.group.x.label	**string**	**ch24: Priority, ch25: Emergency, others: Null** **ch1, 24, 25: 1, others: 0**
The label to identify the group		
ptt.pageMode.group.x.subscribed	**0 or 1**	**1**
Subscribe the phone to the group.		

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 is 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 does not subscribe to the group. If enabled, the phone subscribes to the group.

Parameter	Permitted Values	Default
ptt.pageMode.payloadSize	**10, 20, ..., 80 milliseconds**	**20**
The page mode audio payload size.		
ptt.pageMode.priorityGroup	**1 to 25**	**24**
The paging group to use for priority pages.		
ptt.pageMode.transmit.timeout.continuation	**0 to 65535**	**60**
The time (in seconds) to add to the initial timeout (`ptt.pageMode.transmit.timeout.initial`) for terminating page announcements. If this value is non-zero, an **Extend** soft key displays on the phone. Pressing the **Extend** soft key continues the initial timeout for the time specified by this parameter. If 0, announcements cannot be extended.		
ptt.pageMode.transmit.timeout.initial	**0 to 65535**	**0**
The number of seconds to wait before automatically terminating an outgoing page announcement. If 0, page announcements do not automatically terminate.		

Push-To-Talk and Group Paging Parameters (continued)

Parameter	Permitted Values	Default
ptt.pttMode.enable	**0 or 1**	**0**
Enable or disable push-to talk mode.		
ptt.volume	**-57 to 0**	**-20**
Controls the volume level for pages without changing the volume level for incoming calls.		

These parameters listed in the next table configure trol 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 overrides the other `qos.ip.callControl.*` parameters. The default value is Null, so the other `qos.ip.callControl.*` parameters are used if no value is entered.		
qos.ip.callControl.max_reliability[1]	**0 or 1**	**0**
qos.ip.callControl.max_throughput[1]	**0 or 1**	**0**
qos.ip.callControl.min_cost[1]	**0 or 1**	**0**
qos.ip.callControl.min_delay[1]	**0 or 1**	**1**
qos.ip.callControl.precedence[1]	**0 -7**	**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.		

Quality of Service (Type-of-Service) Parameters (continued)

Parameter	Permitted Values	Default
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 overrides the other `qos.ip.rtp.*` parameters. The default value is Null, so the other `quality.ip.rtp.*` parameters are used.

Parameter	Permitted Values	Default
qos.ip.rtp.max_reliability[1]	0 or 1	0
qos.ip.rtp.max_throughput[1]	0 or 1	1
qos.ip.rtp.min_cost[1]	0 or 1	0
qos.ip.rtp.min_delay[1]	0 or 1	1
qos.ip.rtp.precedence[1]	0 -7	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 overrides the other `qos.ip.rtp.video.*` parameters. The default value is Null, so the other `qos.ip.rtp.video.*` parameters are used.

Parameter	Permitted Values	Default
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.

This section lists all per-registration parameters you can configure. Per-registration parameters apply to a single unique registered line on a phone. You also have the option of associating each registration with a private array of servers for segregated signaling. To see the maximum number of registered lines all Polycom phones support see the table Flexible Call Appearances.

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 is 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 or the H.323 ID/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 applies the dial plan locally. If 0, the dialplan from the Microsoft Lync Server is not used.		
reg.x.advancedConference.maxParticipants	**0-25**	**3**
Sets the maximum number of participants allowed in a push to conference for advanced conference calls. The number of participants configured must match the number of participants allowed on the ALU CTS.		
reg.x.advancedConference.pushToConference	**0 or 1**	**0**
Enable or disable push-to-conference functionality.for advanced conferences for ALU. If enabled, users can select multiple contacts when initiating a conference and during an active conference. Note: The values for this parameter must match does configured on the server.		
reg.x.advancedConference.subscribeForConfEvent s	**0 or 1**	**1**
Enable or disable conference participants to receive notifications for conference events.		
reg.x.advancedConference.subscribeForConfEvent sOnCCPE	**0 or 1**	**1**
Enable or disable the conference host to receive notifications for conference events.		
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 overrides the password entered into the Authentication submenu on the Settings menu of the phone.		
reg.x.auth.userId	**string**	**Null**

Registration Parameters (continued)

Parameter	Permitted Values	Default
User ID to be used for authentication challenges for this registration. If the User ID is non-Null, it overrides 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.bridgeInEnabled	**0 or 1**	**0**
Enable or disable the Bridge In feature.		
reg.x.broadsoft.userId	**String**	**Null**
Enter the BroadSoft user ID to authenticate with the BroadSoft XSP service interface.		
reg.x.broadsoft.useXspCredentials	**0 or 1**	**1**
Set to 0 if registering lines with a server running BroadWorks R19 SP1 or later. Set to 1 if registering lines with a server running BroadWorks R19 or earlier. If this parameter is disabled, the phones use standard SIP credentials to authenticate.		
reg.x.broadsoft.xsp.password	**String**	**Null**
Enter the password associated with the BroadSoft user account for the line. Required only when `reg.x.broadsoft.useXspCredentials=1`.		
reg.x.callsPerLineKey[1]	**1-8, 1-24**	**24 (for VVX phones)**
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.displayName	**UTF-8 encoded string**	**Null**
The display name used in SIP signaling and/or the H.323 alias used as the default caller ID.		
reg.x.enablePvtHoldSoftKey	**0 or 1**	**0**
Enable or disable the Private Hold soft key for a specific line. Set to 1 to display the PvtHold soft key. This parameter applies only to shared lines.		
reg.x.enhancedCallPark.enabled	**0 or 1**	**0**
Enables and disables the BroadWorks Enhanced Call Park feature.		

Registration Parameters (continued)

Parameter	*Permitted Values*	*Default*
reg.x.filterReflectedBlaDialogs	**0 or 1**	**1**

If 0, bridged line appearance NOTIFY messages (dialog state change) is not ignored. If 1, the messages are 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` is used.

reg.x.fwd.busy.status	**0 or 1**	**0**

If 0, incoming calls that receive a busy signal is not 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` is 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.gruu	**0 or 1**	**0**

Specify if the phone sends sip.instance in the REGISTER request.

reg.x.label	**UTF-8 encoded string**	**Null**

The text label that displays next to the line key for registration x.
If Null, the label is determined as follows:
- If `reg.1.useteluriAsLineLabel=1`, then the tel URI/phone number/address displays as the label.
- If `reg.1.useteluriAsLineLabel=0`, then the value for `reg.x.displayName`, if available, displays as the label. If `reg.x.displayName` is unavailable, the user part of `reg.x.address` is used.

Note that the maximum number of characters for this parameter value is 256; however, the maximum number of characters that a phone can display on its user interface varies by phone model and by the width of the characters you use. Parameter values that exceed the phone's maximum display length are truncated by ellipses (…). The rules for parameter up.cfgLabelElide determine how the label is truncated.

reg.x.lineAddress	**String**	**Null**

The line extension for a shared line. If there is no extension provided for this parameter, the call park notification is ignored for the shared line.
Note: This parameter does not apply for private lines.

reg.x.lineKeys	**1 to max**	**1**

Registration Parameters (continued)

Parameter	Permitted Values	Default

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.

---

reg.x.line.y.label

Configure a unique line label for a shared line that has multiple line key appearances. This parameter takes effect when `up.cfgUniqueLineLabel=1`. Note that if `reg.x.linekeys=1`, this parameter does not have any effect.

x = the registration index number starting from 1.

Y = the line index from 1 to the value set by `reg.x.linekeys`. Specifying a string sets the label used for the line key registration on phones with multiple line keys.

If no parameter value is set for `reg.x.line.y.label`, the phone automatically numbers multiple lines by prepending "<y>_" where <y> is the line index from 1 to the value set by `reg.x.linekeys`.

- The following examples show labels for line 1 on a phone with user registration 1234, where `reg.x.linekeys=2`:
  - ⌖ If no label is configured for registration, the labels are "1_1234" and "2_1234".
  - ⌖ If `reg.1.line.1.label=Polycom` and `reg.1.line.2.label=VVX`, the labels display as 'Polycom' and 'VVX'.

---

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."

---

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`.

---

reg.x.offerFullCodecListUponResume	**0 or 1**	**1**

1 (default) - The phone sends full audio and video capabilities after resuming a held call irrespective of the audio and video capabilities negotiated at the initial call answer.

0 - The phone does not send full audio and video capabilities after resuming a held call.

---

reg.x.outboundProxy.address	**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**	**duration**

Registration Parameters (continued)

Parameter	Permitted Values	Default
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 does not fail back until a failover event occurs with the current server.		
reg.x.outboundProxy.failOver.failRegistrationOn	**0 or 1**	**1**
When set to 1, and the reRegisterOn parameter is enabled, the phone silently invalidates an existing registration (if it exists), at the point of failing over. When set to 0, and the reRegisterOn parameter is enabled, existing registrations remain active. This means that the phone attempts 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 ends. No SIP messages are sent to the unregistered server. When set to 0, and the reRegisterOn and failRegistrationOn parameters are enabled, signaling is accepted from and sent to a server that h as 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 attempts 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 proceeds with the secondary server. When set to 0, the phone won't attempt to register with the secondary server, since the phone assumes 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.		
reg.x.outboundProxy.transport	**DNSnaptr, TCPpreferred, UDPOnly, TLS, TCPOnly**	**DNSnaptr**

Registration Parameters (continued)

Parameter	Permitted Values	Default
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 is 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.protocol.H323	**0 or 1**	**0**
You can use this parameter for the VVX 500/501, 600/601, and 1500. If 0, H.323 signaling is not enabled for registration x. If 1, H.323 signaling is enabled.		
reg.x.protocol.SIP	**0 or 1**	**1**
You can use this parameter for the VVX 500/501, 600/601, and 1500. If 0, SIP signaling is not enabled for this registration. If 1, SIP signaling is enabled.		
reg.x.proxyRequire	**string**	**Null**
The string that needs to be entered in the Proxy-Require header. If Null, no Proxy-Require is 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. If you use the configuration parameters ringer13 and ringer14 on a single registered line, the phone plays SystemRing.wav.		
reg.x.serverFeatureControl.callRecording	**0 or 1**	**1**
Enable or disable BroadSoft BroadWorks v20 call recording feature for individual phone lines. This per-line parameter overrides values you set for the parameter `voIpProt.SIP.serverFeatureControl.callRecording` which sets the feature for all lines on a phone.		
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 does not perform local Call Forward behavior. If set to 1, the phone performs local Call Forward behavior on all calls received. This parameter overrides `voIpProt.SIP.serverFeatureControl.localProcessing.cf`.		
reg.x.serverFeatureControl.localProcessing.dnd	**0 or 1**	**1**

Registration Parameters (continued)

Parameter	*Permitted Values*	*Default*
If 0 and `reg.x.serverFeatureControl.dnd` is set to 1, the phone does not perform local DND call behavior. If set to 1, the phone performs local DND call behavior on all calls received. This parameter overrides `voIpProt.SIP.serverFeatureControl.localProcessing.dnd`.		
reg.x.serverFeatureControl.securityClassification	**0 or 1**	**0**
Enable or disable the visual security classification feature for a specific phone line.		
reg.x.serverFeatureControl.signalingMethod	**string**	**serviceMsForwardContact**
Controls the method used to perform call forwarding requests to 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 is 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` is also 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 attempt 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.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.		
reg.x.terminationType	**VVX, DECT, or VVX-DECT**	**NULL**
Determines the type of termination that is used for the line where the line can be managed automatically on the VVX, the wireless handset, or on both. X=each registration index.		

Registration Parameters (continued)

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.		
reg.x.useCompleteUriForRetrieve	**0 or 1**	**1**
This parameters overrides `voipPort.SIP.useCompleteUriForRetrieve`. If set to 1, the target URI in BLF signaling uses the complete address as provided in the xml dialog document.		
If set to 0, only the user portion of the XML dialog document is used and the current registrar's domain is appended to create the full target URI.		

[1] Change causes phone to restart or reboot.

You can list multiple registration servers for fault tolerance. The next table shows how you can list up to four servers by using y=1 to 4. If `reg.x.server.y.address` is not null, all of the parameters in the following table override the parameters specified in `voIpProt.server.*`.

Registration Server Parameters

Parameter	Permitted Values	Default
regOnPhone	**0 or 1**	**Lync = 0**
Enables and disables all line keys. If 1 (enabled), the line keys are moved to the expansion module if one is connected.		
reg.x.server.H323.y.address	**IP address or hostname**	**Null**
Address of the H.323 gatekeeper.		
reg.x.server.H323.y.port	**0 to 65535**	**0**
Port to be used for H.323 signaling. If set to Null, 1719 (H.323 RAS signaling) is used.		
reg.x.server.H323.y.expires	**positive integer**	**3600**
Desired registration period.		
reg.x.server.y.address	**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 override the parameters specified in `voIpProt.server.*`. Notes: If this parameter is set, it takes 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 attempts to re-register at the beginning of the overlap period. For example, if `expires="300"` and `overlap="5"`, the phone re-registers after 295 seconds (300–5).		
reg.x.server.y.expires.lineSeize	**0 to 65535**	**30**

Registration Server Parameters (continued)

Parameter	Permitted Values	Default
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 tries to re-register at half the expiration time returned by the server if the server value is less than the configured overlap value.		
reg.x.server.y.failOver.failBack.mode	**newRequests DNSTTL registration duration**	**duration**

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 does not fail back until a failover 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 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 remain active. This means that the phone attempts 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 ends. No SIP messages are sent to the unregistered server. When set to 0, and the reRegisterOn and failRegistrationOn parameters are enabled, signaling is 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 `voIpProt.server.x.failOver.reRegisterOn`. When set to 1, the phone attempts 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 proceeds with the secondary server. When set to 0, the phone won't attempt to register with the secondary server, since the phone assumes that the primary and secondary servers share registration information.		
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`.		

Registration Server Parameters (continued)

Parameter	Permitted Values	Default
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 *SIP Server Fallback Enhancements on Polycom Phones - Technical Bulletin 5844* on Polycom Engineering Advisories and Technical Notifications.

reg.x.server.y.registerRetry.baseTimeOut	**10 - 120 seconds**	**60**

For registered line x, set y to the base time period the phone waits before trying to re-register with the server. 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 seconds**	**180**

For registered line x, set y to the maximum time period the phone waits before trying to re-register with the server. Use 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 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.subscribe.expires	**10 – 2147483647 seconds**	**3600 seconds**

The phone's requested subscription period in seconds after which the phone attempts to resubscribe at the beginning of the overlap period. You can use this parameter in conjunction with `reg.x.server.y.subscribe.expires.overlap`. For example, if expires="300" and overlap="5", the phone resubscribes after 295 seconds (300–5). Note that the period negotiated with the server may be different.

reg.x.server.y.subscribe.expires.overlap	**5 – 65535 seconds**	**60 seconds**

The number of seconds before the expiration time returned by server x after which the phone attempts to resubscribe. If the server value is less than the configured overlap value, the phone tries to resubscribe at half the expiration time returned by the server.

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 is used.
- **TLS**   If TLS fails, transport fails. Leave port field empty (defaults to `5061`) or set to `5061`.
- **TCPOnly**   Only TCP is used.

Registration Server Parameters (continued)

Parameter	Permitted Values	Default
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.

The parameters listed in the following table configure 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 is 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.

The phone uses built-in sampled audio files (SAF) in wave file format for some sound effects. You can add files downloaded from the provisioning server or from the Internet. Ringtone files are stored in volatile memory which allows a maximum size of 600 kilobytes (614400 bytes) for all ringtones.

The phones support the following sampled audio WAVE (.wav) file formats:

- mono 8 kHz G.711 u-Law   Supported on all phones
- mono L16/8000 (16-bit dynamic range, 8-kHz sample rate)   Supported on all phones
- G.711 A-Law   Supported on all phones
- mono 8 kHz A-law/mu-law   Supported on all phones
- L8/16000 (16-bit, 8 kHz sampling rate, mono)   Supported on all phones
- L16/16000 (16-bit, 16 kHz sampling rate, mono)   Supported on all phones
- L16/32000 (16-bit, 32 kHz sampling rate, mono)   Supported on VVX 500/501, 600/601, and 1500
- L16/44100 (16-bit, 44.1 kHz sampling rate, mono)   Supported on VVX 500/501, 600/601, and 1500
- L16/48000 (16-bit, 48 kHz sampling rate, mono)   Supported on VVX 500/501, 600/601, and 1500

In the following table, *x* is the sampled audio file number.

Sampled Audio File Parameters

Parameter	Permitted Values	Default
saf.x	Null, valid path name, or an RFC 1738-compliant URL to an HTTP, FTP, or TFTP wave file resource.	

- If Null, the phone uses a built-in file.
- If set to a path name, the phone attempt
- s to download this file at boot time from the provisioning server.
- If set to a URL, the phone attempt
- s to download this file at boot time from the Internet.

Note: A TFTP URL must be in the format: `tftp://<host>/[pathname]<filename>`, for example: `tftp://somehost.example.com/sounds/example.wav`.

Note that to use a welcome sound you must enable the parameter `up.welcomeSoundEnabled` and specify a file in `saf.x`. The default UC Software welcome sound file is `Welcome.wav`. For information, see the section Customize Audio Sound Effects.

The next table defines the phone's default use of the sampled audio files.

Default Sample Audio File Usage

Sampled Audio File Number	Default Use (Pattern Reference)
1	Ringer 12 (`se.pat.misc.welcome`)
2	Ringer 15 (`se.pat.ringer.ringer15`)
3	Ringer 16 (`se.pat.ringer.ringer16`)
4	Ringer 17 (`se.pat.ringer.ringer17`)
5	Ringer 18 (`se.pat.ringer.ringer18`)
6	Ringer 19 (`se.pat.ringer.ringer19`)
7	Ringer 20 (`se.pat.ringer.ringer20`)
8	Ringer 21 (`se.pat.ringer.ringer21`)
9	Ringer 22 (`se.pat.ringer.ringer22`)
10	Ringer 23 (`se.pat.ringer.ringer23`)
11	Ringer 24 (`se.pat.ringer.ringer24`)
12 to 24	Not Used

The next table lists configurable sound effect parameters. Sound effects are defined by patterns: rudimentary sequences of chord-sets, silence periods, and wave files. You can also configure sound effect patterns in and ringtones in <rt/>. The phone uses both synthesized and sampled audio sound effects.

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 is 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 play from the destination that is currently in use. For example, if you are in a call on the handset, a new incoming call rings 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.

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 next table.

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 following table, x is the pattern name, y is the instruction number, and cat is the sound effect pattern category. Both x and y need to be sequential. There are three categories of sound effect patterns that you can use to replace cat in the parameter names: `callProg` (Call Progress Patterns), `ringer` (Ringer Patterns) and `misc` (Miscellaneous Patterns).

Sound Effects Pattern Parameters

Parameter	Permitted Values
se.pat.callProg.secondaryDialTone.name Found in region.cfg	**1-255**
se.pat.callProg.secondaryDialTone.inst.1.type Found in region.cfg	**0-255**
se.pat.callProg.secondaryDialTone.inst.1.value Found in region.cfg	**0-50**
se.pat.callProg.secondaryDialTone.inst.1.param This is a debug parameter.	
se.pat.callProg.secondaryDialTone.inst.1.atten This is a debug parameter.	
se.pat.callProg.secondaryDialTone.inst.1.atten Sound effects name, where cat is `callProg`, `ringer`, or `misc`.	**UTF-8 encoded string**
se.pat.cat.x.inst.y.type Type of sound effect, where cat is `callProg`, `ringer`, or `misc`.	**sampled, chord, silence, branch**
se.pat.cat.x.inst.y.value 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`.	**String**

The next table lists 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)

Call Progress Tone Pattern Names (continued)

Call Progress Pattern Name	Description
callWaitingRingback	Call Waiting RingBack Tone
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 next table lists 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

Ringtone Pattern Names (continued)

Parameter Name	Ringtone Name	Description
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 provides 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

The next table lists the miscellaneous patterns and their descriptions.

Miscellaneous Pattern Names

Parameter Name	Miscellaneous pattern name	Description
instantmessage	instant message	New instant message
localHoldNotification	local hold notification	Local hold notification
messageWaiting	message waiting	New message waiting indication
negativeConfirm	negative confirmation	Negative confirmation
positiveConfirm	positive confirmation	Positive confirmation
remoteHoldNotification	remote hold notification	Remote hold notification
welcome	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 rings that are defined as follows:

- **Ring**   Plays a specified ring pattern or call waiting indication.
- **Visual**   Provides a visual indication (no audio) of an incoming call;, no ringer needs to be specified.
- **Answer**   Provides auto-answer on an incoming call.
- **Ring-answer**   Provides auto-answer on an incoming call after a certain number of rings.

> **Note: Use the answer ring type**
> The auto answer for an incoming call works only when there is no other call in progress on your phone, including no other calls in progress on phone lines you share or are monitoring. However, if a phone initiates a call on a line you are sharing or monitoring, auto answer on your phone works.

The phone supports the following ring classes:

- default
- visual
- answerMute
- autoAnswer
- ringAnswerMute
- ringAutoAnswer
- internal
- external
- emergency
- precedence
- splash
- custom*<y>* where y is 1 to 17.

In the following table, x is the ring class name.

> **Caution: Ringtone parameters do not work after a software downgrade**
> If you are using Polycom UC Software 4.0.0 or later and then downgrade to SIP 3.2.3 or earlier, the ringtone parameters are unusable due to configuration parameters name changes in UC Software 4.0.0.

Sound Effects Ringtone Parameters

Parameter	Permitted Values	Default Value
se.rt.enabled	**0 or 1**	**1**
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**	**1**

Sound Effects Ringtone Parameters (continued)

Parameter	Permitted Values	Default Value
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 the table 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 must match one in the table 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.		

The parameters listed in the next table configure security features of the phone.

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:

-
-
-
- <H235/>
- <dot1x>
-

- <TLS/>

The next table 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.

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/>

The next table 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.

Password Length Parameters (continued)

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.

As per RFC 3711, you cannot turn off authentication of RTCP. The next table 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 is 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**	**Lync = 1 Generic = 0**
If enabled, the phone sends two encrypted attributes in the SDP, one with MKI and one without MKI. If disabled, the phone sends only one encrypted attributed without MKI.		
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 increments 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 is included in offered SDP. If 0, the crypto line is not included.		
sec.srtp.offer.HMAC_SHA1_80[1]	**0 or 1**	**1**

SRTP Parameters (continued)

Parameter	Permitted values	Defaults
If 1, a crypto line with the `AES_CM_128_HMAC_SHA1_80` crypto-suite is included in offered SDP. If 0, the crypto line is not included.		
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 is 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` is also 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` is also 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` is also 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 is also set to 1, regardless of the value in the configuration file.		
sec.srtp.simplifiedBestEffort	**0 or 1**	**0**

SRTP Parameters (continued)

Parameter	Permitted values	Defaults
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.

<H235/>

You can use the parameters listed in the next table with the Polycom VVX 500/501, 600/601, and 1500 business media phones. The H.235 Voice Profile implementation is Polycom HDX compatible. OpenSSL-based Diffie-Hellman key exchange and AES-128 CBC encryption algorithms are used to encrypt the RTP media.

H.235 Media Encryption Parameters

Parameter	Permitted Values	Default
sec.H235.mediaEncryption.enabled[1]	**0 or 1**	**1**
If 0, H.235 Voice Profile RTP media encryption is disabled. If 1, H.235 media encryption is enabled and negotiated when such encryption is requested by the far end.		
sec.H235.mediaEncryption.offer[1]	**0 or 1**	**0**
If 0, media encryption negotiations is not initiated with the far end. If 1 and `sec.H235.mediaEncryption.enabled` is also 1, media encryption negotiations is initiated with the far end; however, successful negotiations are not a requirement for the call to complete.		
sec.H235.mediaEncryption.require[1]	**0 or 1**	**0**
If 0, media encryption negotiations are not required. If 1 and `sec.H235.mediaEncryption.enabled` is also 1, media encryption negotiations are initiated or completed with the far end, and if negotiations fail, the call is dropped.		

[1] Change causes phone to restart or reboot.

<dot1x>

The next table 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 does not send an EAPOL Logoff message on behalf of the disconnected supplicant. If 1, the feature is enabled and the phone sends 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**

802.1X EAP over LAN (EAPOL) Logoff Parameters (continued)

Parameter	Permitted Values	Default
If 0, the phone software does not reset (recycle) the LAN port link in the application initiation stage. If 1, the LAN port link resets in the application initiation stage.		

[1] Change causes phone to restart or reboot.

The next table 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 unconditionally sends 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, there is an x microsecond time interval between two consecutive link–up state change reports, which reduces the frequency of dispatching CDP packets.		

[1] Change causes phone to restart or reboot.

<TLS/>

The next table lists configurable TLS parameters. For the list of configurable ciphers, refer to the table Configurable TLS Cipher Suites.

This parameter also includes:

-
- .

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 at: http://www.openssl.org/docs/apps/ciphers.html.		
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 at: http://www.openssl.org/docs/apps/ciphers.html.		
sec.TLS.customCaCert.x	**String**	**Null**

TLS Parameters (continued)

Parameter	Permitted Values	Default
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: http://www.openssl.org/docs/apps/ciphers.html.		
sec.TLS.LDAP.strictCertCommonNameValidation	**0 or 1**	**1**
Specifies whether the server certificate common name must be validated during an LDAP or LDAPS connection over TLS. 1 (default) - Requires validation of server certificate common name during LDAP or LDAPS connection over TLS. 0 - Does not require validation of server certificate common name during LDAP or LDAPS connection over TLS.		
sec.TLS.profileSelection.SOPI	**1 - 7**	**PlatformProfile1**
Select the platform profile you want to use. You can choose platform profile 1 - 7.		
sec.TLS.profile.webServer.cipherSuiteDefault	**0 or 1**	**1**
If 0, use the custom cipher suite for web server profile. If 1, use the default cipher suite.		
sec.TLS.profile.x.caCert.application7	**0 or 1**	**1**
Enable or disable the ability to choose a CA certificate for the application7 profile.		
sec.TLS.profile.webServer.cipherSuiteDefault	**0 or 1**	**1**
If 0, use the custom cipher suite for web server profile. If 1, use the default cipher suite.		
sec.TLS.prov.cipherList	**String**	**NoCipher**
The cipher list for provisioning. The format for the cipher list uses OpenSSL syntax found here: http://www.openssl.org/docs/apps/ciphers.html.		
sec.TLS.SIP.cipherList	**String**	**NoCipher**
The cipher list for SIP. The format for the cipher list uses OpenSSL syntax found here: http://www.openssl.org/docs/apps/ciphers.html.		
sec.TLS.SIP.strictCertCommonNameValidation	**0 or 1**	**1**
If 1, enable common name validation for SIP.		

TLS Parameters (continued)

Parameter	Permitted Values	Default
sec.TLS.SOPI.cipherList	**1 – 1024 character string**	**NoCipher**
Choose a cipher key.		
sec.TLS.SOPI.strictCertCommonNameValidation	**0 or 1**	**1**
Enable or disable strict common name validation for the URL provided by the server.		
sec.TLS.syslog.cipherList	**String**	**NoCipher**
The cipher list for syslog. The format for the cipher list uses OpenSSL syntax found here: http://www.openssl.org/docs/apps/ciphers.html.		
sec.TLS.webServer.cipherList	**String (1 - 1024 characters)**	**RSA:!EXP:!LOW:!NULL:!MD5:!RC4:@STRENGTH**
The cipher list for a web server profile.		
sec.TLS.webServer.cipherList	**String**	**RSA:!EXP:!LOW:!NULL:!MD5:!RC4:@STRENGTH**
The cipher list for a web server profile. The format for the cipher list uses OpenSSL syntax found at: http://www.openssl.org/docs/apps/ciphers.html.		

\<profile/\>

Profiles are a collection of related security parameters. The next table 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.application7 **Application CA 7**	**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 7). If set to 0, the CA is not used. If set to 1, the CA is used.		
sec.TLS.profile.x.caCert.defaultList	**String**	**Null**
The list of default CA certificates for TLS Application Profile x (x= 1 to 7).		
sec.TLS.profile.x.cipherSuite	**String**	**Null**
The cipher suite for TLS Application Profile x (where x is 1 to 8).		
sec.TLS.profile.x.cipherSuiteDefault	**0 or 1**	**1**
If 0, use the custom cipher suite for TLS Application Profile x (x= 1 to 8). If 1, use the default cipher suite.		
sec.TLS.profile.x.deviceCert	**Polycom, Platform1, Platform2, Application1, Application2, Application3, Application4, Application5, Application6, Application7**	**Polycom**
The device certificate to use for TLS Application Profile x (x = 1 to 7).		

You can configure the parameters listed in the next table 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
- ApplicationProfile7

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.SOPI	**a TLS profile**	**PlatformProfile1**
The TLS platform profile or TLS application profile (see preceding list) to use for the GENBAND "Subscriber Open Provisioning Interface" (SOPI).		

You can use the softkey parameters 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.

In the following table listing soft key configuration parameters, 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**
This parameter applies only to VVX 300 and 400 series business media phones. Controls 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.		

Soft Key Customization Parameters (continued)

Parameter	Permitted Values	Default
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 , the default is 0.		
softkey.feature.directories[1]	**0 or 1**	**0**
If 1, the **Dir** soft key displays on all platforms. If 0, the **Dir** soft key is disabled for all platforms. The default value is 0.		
softkey.feature.doNotDisturb	**0 or 1**	**1**
Enable or disable the DND soft key on the phone.		
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 1, the Forward soft key displays. If 0, the Forward soft key does not display.		
softkey.feature.intercom	**0 or 1**	**1**
Enable or disable the intercom soft key.		
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.redial	**0 or 1**	**0**
Enables or disables the display of the Redial soft key on the Home screen. **Note**: The parameter feature.enhancedFeatureKeys.enabled must be set to 1 first to configure this feature, and the parameter efk.softkey.alignleft must be set to 1 to move enabled soft keys into the positions of disabled soft keys.		
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 Understand Macro Definitions.		

Soft Key Customization Parameters (continued)

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.

Parameter	Permitted Values	Default
softkey.x.insert	**0 to 10**	**0**

The position on the phone screen for soft key x. If 0, the phone places the soft key x in the first available position. If 1 or above, the phone places the soft key x in the corresponding position and moves the soft key in that position and the following soft keys one position over to the right.

For example, if usoft.1.insert is set to 3, the soft key is displayed on the screen in the third position from the left. If a soft key was already in the third position, that soft key is moved to the fourth position, and the following soft keys are moved to the right by one space

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 is positioned last after the other soft keys.

Parameter	Permitted Values	Default
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 is used.
- If the soft key calls a speed dial, the label of the speed dial contact is 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 is blank.

Note that the maximum number of characters for this parameter value is 16; however, the maximum number of characters that a phone can display on its user interface varies by phone model and by the width of the characters used. Parameter values that exceed the phone's maximum display length are truncated by ellipses (…). The phone truncates the beginning of numerical labels (for example, …4567) and truncates the end of alphabetical labels (for example, Abcd…).

Parameter	Permitted Values	Default
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.

Parameter	Permitted Values	Default
softkey.x.use.active	**0 or 1**	**0**
Display in the active call state		
softkey.x.use.alerting	**0 or 1**	**0**
Display in the alerting state		
softkey.x.use.dialtone	**0 or 1**	**0**
Display in the dial tone state		
softkey.x.use.hold	**0 or 1**	**0**
Display in the hold state		
softkey.x.use.idle	**0 or 1**	**0**
Display in the idle state		
softkey.x.use.park		
Display in the parked state		
softkey.x.use.proceeding	**0 or 1**	**0**
Display in the proceeding state		
softkey.x.use.setup	**0 or 1**	**0**
Display in the proceeding state		

Soft Key Customization Parameters (continued)

Parameter	Permitted Values	Default
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.		

[1] Change causes phone to restart or reboot.

This parameter includes:

-
-
-
-
-
-
-

The DHCP parameters listed in the next table enable you to configure 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.		

The parameters listed in the next table enables you to set Domain Name System (DNS). However, any values set through DHCP have a higher priority and any values set through the parameter in a configuration file have a lower priority.

Domain Name System (DNS) Parameters

Parameter	Permitted Values	Default
tcpIpApp.dns.address.overrideDHCP[1]	**0 or 1**	**0**
If set to 0, a DNS address is requested from the DHCP server. When set to 1, a DNS primary and secondary address is set using the parameters `tcpIpApp.dns.server` and `tcpIpApp.dns.altServer`.		
tcpIpApp.dns.server[1]	**IP address**	**Null**
The primary server to which the phone directs DNS queries.		

Domain Name System (DNS) Parameters (continued)

tcpIpApp.dns.altServer[1]	IP address	Null
The secondary server to which the phone directs DNS queries.		
tcpIpApp.dns.domain[1]	String	Null
The phone's DNS domain.		
tcpIpApp.dns.domain.overrideDHCP[1]	0 or 1	0
If set to 0, a domain name is retrieved from the DHCP server, if one is available. If set to 1, the DNS domain name is set using the parameter `tcpIpApp.dns.domain`.		

[1] Change causes phone to restart or reboot.

Parameters in the following table enable you to set the STUN/TURN/ICE feature.

ICE Parameters

Parameter	Permitted Values	Default
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.		

The next table lists the Simple Network Time Protocol (SNTP) parameters used to set up time synchronization and daylight savings time. The default values 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.AQuery	**0 or 1**	**0**
If set to 0, queries to resolve the SNTP hostname are performed using DNS SRV. If set to 1, the host name is queried for a DNS A record instead.		
tcpIpApp.sntp.address.overrideDHCP	**0 or 1**	**0**
If 0, the DHCP values for the SNTP server address are used. If 1, the SNTP parameters 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.		

Simple Network Time Protocol (SNTP) Parameters (continued)

Parameter	Permitted Values	Default
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.

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.gmtOffsetcityID	**0 - 127**	**Null**

For descriptions of all values, refer to Set Time Zone Location Description.

tcpIpApp.sntp.gmtOffset.overrideDHCP	**0 or 1**	**0**

If 0, the DHCP values for the GMT offset are used. If 1, the SNTP values for the GMT offset are 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.

tcpIpApp.sntp.retryDnsPeriod	**60 – 2147483647 seconds**	**86400**

Set a retry period for DNS queries. Note that the DNS retry period you configure is affected by other DNS queries made by the phone. If the phone makes a query for another service such as SIP registration during the retry period you configure and receives no response, the Network Time Protocol (NTP) DNS query is omitted to limit the overall number of retry attempts made to the unresponsive server. If no other DNS attempts are made by other services, then the rety period you configure is not affected. If at any time the DNS server becomes responsive to another service, then NTP also immediately retries its DNS query as well.

<port/>

The parameters listed in the next table enable you to configure the port filtering used for RTP traffic.

RTP Port Parameters

Parameter	Permitted Values	Default
tcpIpApp.port.rtp.filterByIp[1]	**0 or 1**	**1**
IP addresses can be negotiated through the SDP or H.323 protocols. If set to 1, the phone rejects RTP packets that arrive from non-negotiated IP addresses. Note that the H.323 protocol is supported on the VVX 500/501, 600/601, and 1500 phones.		
tcpIpApp.port.rtp.filterByPort[1]	**0 or 1**	**0**
Ports can be negotiated through the SDP protocol. If set to 1, the phone rejects 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.		
tcpIpApp.port.rtp.mediaPortRangeEnd[1]	**Default, 1024 to 65485**	**2269**
Determines the maximum supported end range of audio ports.		
tcpIpApp.port.rtp.mediaPortRangeStart[1]	**even integer 1024 to 65440**	**2222**
Set the starting port for RTP port range packets. Each call increments the port number +2 to a maximum of 24 calls after which the value resets to the starting point. Because port 5060 is used for SIP signaling, ensure that port 5060 does not fall within this range when setting this parameter. A call that tries to use port 5060 will have no audio.		
tcpIpApp.port.rtp.videoPortRange.enable	**0 or 1**	**Base profile** **Lync = 1** **Generic = 0**
If 1, video ports are chosen from the range specified by **tcpIpApp.port.rtp.videoPortRangeStart** and **tcpIpApp.port.rtp.videoPortRangeEnd**. If 0, video ports are also chosen within the range specified by **tcpIpApp.port.rtp.mediaPortRangeStart** and **tcpIpApp.port.rtp.mediaPortRangeEnd**.		
tcpIpApp.port.rtp.videoPortRangeEnd[1]	**Default, 1024 to 65535**	**2319**
Determines the maximum supported end range of video ports.		
tcpIpApp.port.rtp.videoPortRangeStart[1]	**Default, 1024 to 65486**	**2272**
Determines the start range for video ports. This is used only when the value of **tcpIpApp.port.rtp.videoPortRange.enable** is **1**.		

[1] Change causes phone to restart or reboot.

The parameters listed in the next table 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.

Note: On VVX phones and SoundStructure VoIP Interface, this parameter specifies the number of seconds TCP waits between transmission of the last data packet and the first keep-alive message.

Parameter	Permitted Values	Default
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).

Note: On VVX phones and SoundStructure VoIP Interface, this parameter specifies the amount of idle time between the transmission of the keep-alive packets the TCP stack waits. This applies whether the last keep-alive was acknowledged or not.

Parameter	Permitted Values	Default
tcpIpApp.keepalive.tcp.sip.persistentConnection.enable[1]	**0 or 1**	**0**

If 0, the TCP socket opens a new connection when the phone tries to send any new SIP message and closes after one minute. If 1, the TCP socket connection remains open indefinitely.

Parameter	Permitted Values	Default
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.

The parameters listed in the next table configure 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.

This parameter lists configuration items for available tone resources and includes:

- <DTMF/>
-

<DTMF/>

The parameters listed in the next table 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 play through the speakerphone in handsfree mode. If 1 (set only if `tone.dtmf.viaRtp` is set to 0), DTMF tones are 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 is 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 pauses 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 play for. This is also the minimum time the tone plays when dialing manually (even if key press is shorter).		
tone.dtmf.rfc2833Control[1]	**0 or 1**	**1**
If set to 1, the phone indicates 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 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**	**Lync = 101** **Generic = 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.		

[1] Change causes phone to restart or reboot.

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 next table.

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.

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.

Use the parameters listed in the next table 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.accessibilityFeatures	**0 or 1**	**0**
VVX 1500 only. If 0, accessibility features are disabled. If 1, the screen background flashes orange for incoming calls.		
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**
Specify whether you want to use the handset or headset for audio.		
up.backlight.idleIntensity	**VVX 300/301/310/311 = 0, 1, 2, 3** **All other phones = 1, 2, 3**	**1**
The brightness of the LCD backlight when the phone is idle. 1 – low, 2 – medium, and 3 – high. Note: If this is higher than the active backlight brightness (`onIntensity`), the active backlight brightness is used.		
up.backlight.onIntensity	**VVX 300/301/310/311 = 0, 1, 2, 3** **All other phones = 1, 2, 3**	**3**
The brightness of the LCD backlight when the phone is active (in use). 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.basicSettingsPasswordEnabled	**0 or 1**	**0**
If set to 1, a password is required for access to the Basic settings menu on the phone. If set to 0, no password is required to access the Basic settings menu.		
up.cfgLabelElide	**None, Right, Left**	**None**
Controls the alignment of the line label. When the line label is an alphanumeric or alphabetic string, the label aligns right. When the line label is a numeric string, the label aligns left.		
up.cfgUniqueLineLabel	**0 or 1**	**0**
Allow unique labels for the same registration that is split across multiple line keys using reg.X.linekeys. Set to 0 to use the same label on all linekeys. Set to 1 to display a unique label as defined by reg.X.line.Y.label. If reg.X.line.Y.label is not configured, then a label of the form <integer>_ will be applied in front of the applied label automatically.		

User Preferences Parameters (continued)

Parameter	Permitted Values	Default
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 does not display.		
up.echoPasswordDigits	**0 or 1**	**1**
If 1, the phone briefly displays password characters before being masked by an asterisk. If 0, the phone displays only asterisks for the password characters.		
up.em.linkalivecheck.enabled	**0 or 1**	**0**
If 1, the host VVX phone periodically sends ping packets to the expansion modules. If 0, the host VVX phone does not ping the expansion modules.		
up.em.smartpaging.enabled	**0 or 1**	**1**
Enable or disable line key assignments and page distribution on the VVX Expansion Modules. If 1, Smart Paging is enabled. If 0, Smart Paging is disabled. Note that the flexible line key configuration overrides Smart Paging for the expansion module, and Smart Paging is disabled for VVX Expansion Modules with a paper display.		
up.formatPhoneNumbers	**0 or 1**	**1**
Enable or disable automatic number formatting.		
up.handsetModeEnabled	**0 or 1**	**1**
Enable or disable the handset port.		
up.handsfreeMode	**0 or 1**	**1**
If 0, the speakerphone is disabled (cannot be used). If 1, the speakerphone is enabled.		
up.headsetAlwaysUseIntrinsicRinger	**0 or 1**	**1**
If 1, the USB headset uses 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 is used by default instead of the handset. If 1, the headset is used as the preferred audio mode after the headset key is pressed for the first time, until the headset key is pressed again.		
up.headsetModeEnabled	**0 or 1**	**1**
If 0, the headset port is disabled and cannot be used. If 1, the headset port is enabled and can be used.		
up.headset.phoneVolumeControl[1]	**disable, enable, auto**	**auto**

User Preferences Parameters (continued)

Parameter	Permitted Values	Default

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. Note that if `up.prioritizeBackgroundMenuItem.enabled` is 1, you can choose to display the background or the idle browser on 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.IdleViewPreferenceRemoteCalls[1]	**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, for example, on a BLF monitored line, 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, for example, on a BLF monitored line, 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.

up.manualProtocolRouting	**0 or 1**	**1**

You can use this parameter with the VVX 500/501, 600/601, and 1500 phones. If 1, the user is presented with a protocol routing choice in situations where a call can be placed using either protocol (for example, with SIP and H.323 protocols). If 0, the default protocol is used and the user does not choose.

up.manualProtocolRouting.softKeys	**0 or 1**	**1**

User Preferences Parameters (continued)

Parameter	Permitted Values	Default
You can use this parameter with the VVX 500/501, 600/601, and 1500 phones. Choose whether or not you want to display soft keys that control Manual Protocol Routing options. When Soft Key Control is enabled, you can use soft keys to choose between the SIP or H.323 protocol. When disabled, soft keys for protocol routing do not display. The soft keys are enabled by 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 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 shows the caller's name first. If 1, the caller's phone number is shown first.		
up.numOfDisplayColumns[1]	**1, 2, 3, 4**	**VVX 500/501=max 3** **VVX 600/601=max 4**
Set the maximum number of columns the VVX 500/501 and 600/601 display. Note that phones display one column when the value is set to 0. The maximum number of columns for the VVX 500/501 is 3. The maximum number of columns for the VVX 600/601 is 4.		
up.offHookAction.none[1]	**0 or 1**	**0**
If 1, when the user lifts the handset, the phone does not seize the line and the ringer continues until the user takes further action.		
up.OffHookIdleBrowserView.enabled	**0 or 1**	**0**
0 (Default) - The idle browser does not display on screen after the phone goes off hook. 1 - The idle browser continues to display on screen after the phone goes off hook.		
up.oneTouchVoiceMail[1]	**0 or 1**	**0**
If 1, the phone dials voicemail services directly, if available on the call server, without displaying the voicemail summary. If 0, the phone displays a summary page with message counts. Users must press the Connect soft key to dial the voicemail server.		
up.osdIncomingCall.Enabled	**0 or 1**	**1**
If 1, the full screen popup or OSD for incoming calls displays. If 0, the full screen popup or OSD for incoming calls does not display.		
up.pictureFrame.timePerImage	**3 to 300 seconds**	**5**
For the VVX 500/501, 600/601, and 1500 only. The number of seconds to display each picture frame image.		
up.pictureFrame.folder	**string**	**Null**
For the VVX 500/501, 600/601, and 1500 only. The path name for images. The maximum length is 40 characters. If set to Null, images stored in the root folder on the USB flash drive are displayed. For example, if the images are stored in the /images/phone folder on the USB flash drive, set this parameter to `images/phone`.		

User Preferences Parameters (continued)

Parameter	Permitted Values	Default
up.prioritizeBackgroundMenuItem.enabled[1]	**0 or 1**	**1**

If `up.idleBrowser.enabled` is 1, this parameter can be set to 1 to display a **Prioritize Background** menu to the user. The user can choose whether the phone background should take priority over the idle browser or not.

up.ringer.minimumVolume	**0 - 15**	**16**

Configure the minimum ringer volume. This parameter defines how many volume steps are accessible below the maximum level by the user.

16 (default) - The full 16 steps of volume range are accessible.

1-15

0 - Ring volume is not adjustable by the user and the phone uses maximum ring volume.

Upon bootup, the volume is set to ½ the number of configured steps below the maximum (16). So, if the parameter is set to 8, on bootup, the ringer volume is set to 4 steps below maximum.

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 cycles 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 or 2**	**0**

Choose the type of screen saver to display. If 0, the phone screen saver displays default images. If 2, the phone screen saver displays the idle browser. You can use this parameter with the VVX 300 and 400 series phones.

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.SLA.ringType	**default, ringer1 to ringer24**	**ringer2**

Specifies a ring type for Shared Line Appearance (SLA) lines.

up.status.message.flash.rate	**2 - 8 seconds**	**2 seconds**

Controls the scroll rate of the status bar on VVX 300 and 400 series business media phones.

up.transparentLines	**0 or 1**	**0**

User Preferences Parameters (continued)

Parameter	*Permitted Values*	*Default*
If 0, line keys block display of the background image. If 1, line keys are transparent and allow the background image to display behind the line labels. This parameter applies only to the VVX 500/501 and 600/601 business media phones.		
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 is used as the caller ID for incoming calls from contacts in the local directory. Note: Outgoing calls and corporate directory entries are not matched.		
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. All warnings are listed in the Warnings menu. If 2, the phone displays a warning icon and no warning messages.		
For all the values, all warnings are listed in the warning menu. Access to the Warnings menu varies by phone model: • **VVX 1500** Menu > Status > Diagnostics > Warnings • **VVX 101, 201, 300/301/310/311, 400/401/410/411, 500/501, and 600/601** Settings > Status > Diagnostics > Warnings		
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. Note that to use a welcome sound you must enable the parameter `up.welcomeSoundEnabled` and specify a file in `saf.x` as shown in the section <saf/>. The default UC Software welcome sound file is `Welcome.wav`. See the example configuration in the section Customize Audio Sound Effects.		
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.

Use the parameters listed in the next table 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 are supported on the VVX 500/501, VVX 600/601, and VVX 1500.

This parameter also includes:

-
-
-

Video Parameters

Parameter	Permitted Values	Default
video.allowWithSource	**0 or 1**	**0**

Restrict when to send video codec negotiation in SDP. Applies only to the VVX 500/501 and VVX 600/601.

video.enable	video.allowWithSource	Camera Attached	Result
0	0	0	no video codecs advertised
0	1	0	no video codecs advertised
1	0	0	video codecs advertised
1	1	0	no video codecs advertised
0	0	1	no video codecs advertised
0	1	1	no video codecs advertised
1	0	1	video codecs advertised
1	1	1	video codecs advertised
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 always displays if available, and far side users can control when to send video.

video.callMode.default	**audio or video**		**audio**

Video Parameters (continued)

When the device uses SIP protocol, this parameter allows users to select the outbound call mode. Note that you must enable `feature.audioVideoToggle.enabled="1"` to apply this parameter. Note that when you set this parameter to 'video', the VVX 500/501 and 600/601 display a Video Mode soft key, and the VVX 1500 displays a video icon.

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**	**0**

If 1, the first I-Frame request uses the method defined by `video.forceRtcpVideoCodecControl` and subsequent requests alternate between RTCP-FB and SIP INFO.

In case of network device problems, you can set the phone to attempt multiple methods of I-frame requests. To set other methods for I-frame requests, refer to the parameter video.forceRtcpVideoCodecControl.

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.

Note: On the VVX 500/501 and 600/601, when you enable video, the G.722.1C codec is disabled.

video.forceRtcpVideoCodecControl[1]	**0 or 1**	**0**

If 1, the phone is forced to send RTCP feedback messages to request fast update I-frames along with SIP INFO messages for all video calls irrespective of a successful SDP negotiation of a=rtcp-fb. If 0, RTCP-FB messages depend on a successful SDP negotiation of a=rtcp-fb and are not used if that negotiation is missing.

For an account of all parameter dependencies when setting I-frame requests, refer to the section Configure I-Frames.

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.

video.iFrame.minPeriod	**1 - 60**	**2**

After sending an I-frame, the phone always waits 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 is 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 has motion or movement. Use `sharpness` or Null if your outgoing video has little or no movement.

Note: If `motion` is not selected, moderate to heavy motion can cause some frames to be dropped.

[1] Change causes phone to restart or reboot.

The settings in the next table 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).		
video.camera.sharpness	**0 to 6**	**3**
Set sharpness level. The value range is from 0 (Lowest) to 6 (Highest).		

The video codecs include:

-
-

The next table lists video codec parameters.

Video Codec Preference Parameters

Parameter	Permitted Values	Default
video.codecPref.H261[1]	**1 to 4**	**4**
video.codecPref.H264[1]		**1**
video.codecPref.H263 1998[1]		**2**
video.codecPref.H263[1]		**3**

Specifies the video codec preferences for the VVX 500/501, 600/601, and 1500 phones. Note that the VVX 500/501 and 600/601 support H.263 and H.264 and do not support H.261 or H.263 1998.

[1] Change causes phone to restart or reboot.

The next table lists settings for a group of low-level video codec parameters. For most use cases, the default values are appropriate. Polycom does not recommend changing the default values unless specifically advised to do so.

Video Profile Parameters

Parameter	Permitted Values	Default
video.profile.H261.annexD[1]	**0 or 1**	**1**

Enable or disable Annex D when negotiating video calls.

Parameter	Permitted Values	Default
video.profile.H261.CifMpi[1]	**1 to 32**	**1**

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'.

Parameter	Permitted Values	Default
video.profile.H261.jitterBufferMax [1]	**(video.profile.H261.jitter BufferMin + 500ms) to 2500ms**	**2000ms**

The largest jitter buffer depth to be supported (in milliseconds). Jitter above this size always causes lost packets. This parameter should be set to the smallest possible value that support the expected network jitter.

Parameter	Permitted Values	Default
video.profile.H261.jitterBufferMin[1]	**33ms to 1000ms**	**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 still continues. 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	Default
video.profile.H261.jitterBufferShri nk[1]	**33ms to 1000ms**	**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).

Parameter	Permitted Values	Default
video.profile.H261.payloadType[1]	**0 to 127**	**31**

RTP payload format type for H261 MIME type.

Video Profile Parameters (continued)

Parameter	Permitted Values	Default
video.profile.H261.QcifMpi[1]	**1 to 32**	**1**
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 to 32**	**1**
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**	**2000ms**
The largest jitter buffer depth to be supported (in milliseconds). Jitter above this size always causes lost packets. This parameter should be set to the smallest possible value that supports the expected network jitter.		
video.profile.H263.jitterBufferMin[1]	**33ms to 1000ms**	**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 still continues. 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.jitterBufferShri nk[1]	**33ms to 1000ms**	**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**	**34**
RTP payload format type for H263 MIME type.		
video.profile.H263.QcifMpi[1]	**1 to 32**	**1**
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 to 32**	**1**
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 or 1**	**0**
Enable or disable Annex F when negotiating video calls.		
video.profile.H2631998.annexI[1]	**0 or 1**	**0**
Enable or disable Annex I when negotiating video calls.		
video.profile.H2631998.annexJ[1]	**0 or 1**	**0**
Enable or disable Annex J when negotiating video calls.		

Video Profile Parameters (continued)

Parameter	Permitted Values	Default
video.profile.H2631998.annexK[1]	0, 1, 2, 3, 4	1
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'.		
video.profile.H2631998.annexN[1]	0, 1, 2, 3, 4	1
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 or 1	0
Enable or disable Annex T when negotiating video calls.		
video.profile.H2631998.CifMpi[1]	1 to 32	1
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.jitterBuffe rMax[1]	(video.profile.H2631998.jitterBuff erMin+ 500ms) to 2500ms	2000ms
The largest jitter buffer depth to be supported (in milliseconds). Jitter above this size always causes lost packets. This parameter should be set to the smallest possible value that supports the expected network jitter.		
video.profile.H2631998.jitterBuffe rMin[1]	33ms to 1000ms	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 still continues. 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.jitterBuffe rShrink[1]	33ms to 1000ms	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.payloadTy pe[1]	96 to 127	96
RTP payload format type for H263-1998/90000 MIME type.		
video.profile.H2631998.QcifMpi[1]	1 to 32	1
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 to 32	1
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 Parameters (continued)

Parameter	Permitted Values	Default
video.profile.H264.jitterBufferMax [1]	(video.profile.H264.jitterBufferMin + 500ms) to 2500ms	2000ms

The largest jitter buffer depth to be supported (in milliseconds). Jitter above this size always causes lost packets. This parameter should be set to the smallest possible value that supports the expected network jitter.

video.profile.H264.jitterBufferMin[1]	33ms to 1000ms	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 still continues. 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.H264.jitterBufferShrink[1]	33ms to 1000ms	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	109

RTP payload format type for H264/90000 MIME type.

video.profile.H264.profileLevel[1]	1, 1b, 1.1, 1.2, 1.3, and 2	1.3

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, and 2. The default level is 1.3.
Note: VVX 500/501 and VVX 600/601 phones support H.264 with a profile level of 2, and VVX 1500 phones support H.264 with a profile level of 1.3.

1 Change causes phone to restart or reboot.

The parameters in the next table configure how the local camera displays on the screen.

Local Camera View Preferences Parameters

Parameters	Permitted Values	Default
video.localCameraView.fullscreen.enabled	0=Disable, 1=Enable	1

Determines whether the local camera view is shown in the full screen layout.
If set to 0, the local camera view is not shown. If set to 1, the local camera view is shown.

video.localCameraView.fullscreen.mode	pip, side-by-side	side-by-side

Determines how the local camera view is shown. If set to pip, the local camera view displays as a picture-in-picture with the far end window.
If set to side-by-side, the local camera view displays side-by-side with the far end window.

The parameters listed in the next table configure phone audio.

This parameter includes:

-
-
-
-
-
-
- <line/>
-
-
-
-
-

Voice Parameters

Parameter	Permitted Values	Default
voice.txEq.hf.preFilter.enable	**0 or 1**	**0**

If 1 and a narrow band codec is in use, such as G.711mu, G.711A, G.729, or iLBC, a 300 Hz high-pass filter is applied to the transmit audio prior to encoding.
Enabling this filter may improve intelligibility to the far end when making narrow band calls through a PSTN gateway in a noisy environment.

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**

Voice Parameters (continued)

Parameter	Permitted Values	Default
If 0, no Tx filtering is performed. If 1, narrowband Tx high pass filter is enabled.		

[1] Change causes phone to restart or reboot.

Use these parameters to enable or disable the acoustic echo cancellation (AEC) function for a specified termination.

Acoustic Echo Canceller (AEC) Parameters

Parameter	*Permitted Values*	*Default*
voice.aec.hf.enable	**0 or 1**	**1**
Enable or disable the handsfree AEC function. Note: Polycom recommends that you do not disable this parameter.		
voice.aec.hs.enable	**0 or 1**	**1**
Enable or disable the handset AEC function.		

Use these parameters to control the speakerphone acoustic echo suppression (AES). These parameters remove residual echo after AEC processing. Because AES depends on AEC, enable AES only when you also enable AEC using `voice.aec.hd.enable`.

Acoustic Echo Suppression Parameters

Parameter	*Permitted Values*	*Default*
voice.aes.hf.enable	**0 or 1**	**1**
Enable or disable the handsfree AES function. Note: Polycom recommends that you do not disable this parameter.		
voice.aes.hs.enable	**0 or 1**	**1**
Enable or disable the handset AES function.		

Use these parameters to configure the addition and volume of comfort noise during conferences.

Comfort Noise Parameters

Parameter	Permitted Values	Default
voice.cn.hf.enable	**0 or 1**	**1**

If 1, comfort noise is added into the Tx path for hands-free operation. This feature should be used only when users at the far end perceive that the phone has gone "dead" after the near end user stops talking.
If 0, no comfort noise is added.

voice.cn.hf.attn	**0 - 90**	**35 (quite loud)**

Attenuation of the inserted comfort noise from full scale in decibels; smaller values insert louder noise. Use this parameter only when `voice.cn.hf.enabled` is 1.

voice.cn.hd.enable	**0 or 1**	**0**

If 1, comfort noise is added into the Tx path for the headset. This feature should be used only when users at the far end perceive that the phone has gone "dead" after the near end user stops talking.
If 0, no comfort noise is added.

voice.cn.hd.attn	**0 - 90**	**30 (quite loud)**

Attenuation of the inserted comfort noise from full scale in decibels; smaller values insert louder noise. Use this parameter only when `voice.cn.hd.enabled` is 1. Default value is 30, which is quite loud.

voice.cn.hs.enable	**0 or 1**	**1**

If 1, comfort noise is added into the Tx path for the handset. This feature should be used only when users at the far end perceive that the phone has gone "dead" after the near end user stops talking.
If 0, no comfort noise is added.

voice.cn.hs.attn	**0 - 90**	**35 (quite loud)**

Attenuation of the inserted comfort noise from full scale in decibels; smaller values insert louder noise. Use this parameter only when `voice.cn.hs.enabled` is 1. Default value is 30, which is quite loud.

voice.vadRxGain	**-20 to +20 dB**	**0**

When in a narrow band call (for example, G.711mu, G.711A, G.729AB, iLBC), the specified gain value in dB is added to the noise level of an incoming VAD or CNG packet. This causes the noise level being synthesized at the local phone to change by the specified amount. This configuration parameter may be useful for tuning VAD or CNG interop in a multi-vendor environment. When tuning in multi-vendor environments, the existing Polycom to Polycom phone behavior can be retained by setting voice.vadTxGain = -voice.vadRxGain.
This parameter is ignored for HD calls.

voice.vadTxGain	**-20 to +20 dB**	**0**

When in a narrow band call (for example, G.711mu, G.711A, G.729AB, iLBC), the specified gain value in dB is added to the noise level of an incoming VAD or CNG packet. This causes the noise level being synthesized at the local phone to change by the specified amount. This configuration parameter may be useful for tuning VAD or CNG interoperability in a multi-vendor environment. When tuning in multi-vendor environments, the existing Polycom to Polycom phone behavior can be retained by setting voice.vadTxGain = -voice.vadRxGain.
This parameter is ignored for HD calls.

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 next table.

If a particular phone does not support a codec, the phone ignores that codec and continue to the codec next in the priority. For example, using the default values, the highest-priority codec on a VVX 310 phone is G.722.1 since that model doesn't support G.722.1C or G.719.

For more information on codecs on particular phones and priorities, see Supported 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

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 treats the setting as if it were 0 and not offer or accept calls with that codec.

The parameters listed in this section control the level of sidetone on handsets of VVX business media phones.

Handset Parameters

Parameter	Permitted Values	Default
voice.handset.st	**-12 to +12**	**0**

Adjust the handset sidetone level from the default in 1 decibel (dB) increments.

The parameters listed in this section control the level of sidetone on headsets connected to VVX business media phones.

Headset Parameters

Parameter	Permitted Values	Default
voice.headset.st	**-12 to +12**	**0**

Adjust the headset sidetone level from the default in 1 decibel (dB) increments.

<line/>

The following parameters control audio level settings for phone handset and headset.

Voice Line Parameters

Parameter	Permitted Values	Default
voice.lineAgc.hs.enable	**0 or 1**	**0**

Enable or disable the line automatic gain control function which automatically boosts the volume of low-level signals and reduces high-level signals on the phone handset. If 0, the line automatic gain control is disabled on the handset. If 1, the line automatic gain control is enabled for the handset. This parameter applies to the VVX 300 series, 400 series, 500 series, and 600 series business media phones.

voice.lineAgc.hd.enable	**0 or 1**	**0**

Enable or disable the line automatic gain control function which automatically boosts the volume of low-level signals and reduces high-level signals on the phone headset. If 0, the line automatic gain control is disabled on the headset. If 1, the line automatic gain control is enabled for the headset. This parameter applies to the VVX 300 series, 400 series, 500 series, and 600 series business media phones.

This section lists noise suppression parameters available with the Acoustic Fence and Polycom NoiseBlock features.

Noise Suppression Parameters

Parameter	Permitted Values	Default
voice.ns.hd.enable	**0 or 1**	**1**
Enable or disable noise suppression for headsets.		
voice.ns.hd.enhanced	**0 or 1**	**0**
Enable or disable Acoustic Fence noise suppression for headsets. Note that voice.ns.hd.enable must also be set to 1 to use this parameter.		
voice.ns.hd.nonStationary Thresh	**1 -10**	**8**
Increase or decrease the Acoustic Fence noise suppression threshold for headsets. A lower value allows more background sound to enter, and a higher value suppresses background noise. Note that high values can suppress the speaker's voice and impact far-end audio quality.		
voice.ns.hf.block	**0 or 1**	**0**
If 0, Polycom NoiseBlock technology is disabled. If 1, NoiseBlock is enabled.		
voice.ns.hs.enable	**0 or 1**	**1**
Enable or disable noise suppression for handsets.		
voice.ns.hs.enhanced	**0 or 1**	**0**
Enable or disable Acoustic Fence noise suppression for handsets. Note that voice.ns.hs.enable must also be set to 1 to use this parameter.		
voice.ns.hs.nonStationary Thresh	**1 -10**	**8**
Increase or decrease the Acoustic Fence noise suppression threshold for handsets. A lower value allows more background sound to enter, and a higher value suppresses background noise. Note that high values can suppress the speaker's voice and impact far-end audio quality.		

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. The next able lists parameters that configure 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.bluetooth.headset	**0 or 1**	**0**
If 0, the Bluetooth headset are not used for every call. If 1, the Bluetooth headset are used for all calls.		

Volume Parameters (continued)

Parameter	Permitted Values	Default
voice.volume.persist.handset	**0 or 1**	**0**
If 0, the handset receive volume automatically resets to a nominal level after each call. If 1, the volume for each call are the same as the previous call. If set to 1, the handset receive volume persists across calls. If set to 0, the handset receive volume resets to nominal at the start of each call.		
voice.volume.persist.headset	**0 or 1**	**0**
If 0, the headset receive volume automatically resets to a nominal level after each call. If 1, the volume for each call is the same as the previous call.		
voice.volume.persist.handsfree	**0 or 1**	**1**
If 0, the speakerphone receive volume automatically resets to a nominal level after each call. If 1, the volume for each call is the same as the previous call.		
voice.volume.persist.usb.handsfree	**0 or 1**	**0**
If 0, the USB headset is not used. If 1, the USB headset is used.		
voice.volume.persist.usbHeadset	**0 or 1**	**0**
If 0, the USB headset is not used. If 1, the USB headset is used.		

The parameters listed in the next table configure voice activity detection (silence suppression) feature.

Voice Activity Detection (VAD) Parameters

Parameter	Permitted Values	Default
voice.CNControl	**0 or 1**	**1**
Publishes support for Comfort Noise in the SDP body of the INVITE message and includes the supported comfort noise payloads in the media line for audio. If set to 1, either the payload type 13 for 8 KHz sample rate audio codec is sent for Comfort Noise, or the dynamic payload type for 16 KHz audio codecs are sent in the SDP body.		
voice.CN16KPayload	**96 to 127**	**122**
Alters the dynamic payload type used for Comfort Noise RTP packets for 16 KHz codecs.		
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).		
voice.vadEnable[1]	**0 or 1**	**1**

Voice Activity Detection (VAD) Parameters (continued)

Parameter	Permitted Values	Default
If 0, voice activity detection (VAD) is disabled. If 1, VAD is enabled.		
voice.vadThresh[1]	**integer from 0 to 30**	**25**
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.

The next table lists Voice Quality Monitoring (VQMon) 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.		
voice.qualityMonitoring.collector.enable.triggeredPeriodic[1]	**0 to 2**	**0**

Voice Quality Monitoring Parameters (continued)

Parameter	Permitted Values	Default
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 90 seconds**	**900 seconds**
The time interval between successive periodic quality reports.		
voice.qualityMonitoring.collector.server.x.address[1]	**IP address or hostname**	**Null**
The server address 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.collector.server.x.outboundProxy.address	**IP address or FQDN**	**NULL**
When configured, this parameter directs SIP messages related to voice quality monitoring to a separate proxy. No failover is supported for this proxy, and voice quality monitoring is not available for error scenarios.		
voice.qualityMonitoring.collector.server.x.outboundProxy.port	**0 to 65535**	**0**
Specify the port to use for the voice quality monitoring outbound proxy server.		
voice.qualityMonitoring.collector.server.x.outboundProxy.transport	**DNSnaptr, TCPpreferred, UDPOnly, TLS, TCPOnly**	**DNSnaptr**
Specify the transport protocol the phone uses to send the voice quality monitoring SIP messages.		
voice.qualityMonitoring.collector.server.x.port[1]	**1 to 65535**	**5060**
The 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.failover.enable	**0 or 1**	**1**
If 1, the phone will perform a failover when voice quality SIP PUBLISH messages are unanswered by the collector server. If 0, no failover is performed; note, however, that a failover is still triggered for all other SIP messages. This parameter is ignored if `voice.qualityMonitoring.collector.server.x.outboundProxy` is enabled.		
voice.qualityMonitoring.location	**Valid location string**	**Unknown**
Specify the device location with a valid location string. If you do not configure a location value, you must use the default string 'Unknown'.		
voice.qualityMonitoring.rfc6035.enable	**0 or 1**	**0**
If 0, the existing draft implementation is supported. If 1, complies with RFC6035.		
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.

The following table lists the jitter buffer parameters for wired 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.**	**0 to 80**	**20**
voice.rxQoS.maxJitter[1] **The maximum expected jitter.**	**0 to 200**	**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.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.

avgJitter   The PTT/Paging interface minimum depth will be automatically configured to adaptively handle this level of continuous jitter without packet loss.

maxJitter   The PTT/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 PTT/Paging interface interfaces.

[1] Change causes phone to restart or reboot.

You must set up the call server and DTMF signaling parameters.

This parameter includes:

-
- <SDP/>
- <SIP/>
- <H323/>

The next table describes VoIP server configuration parameters.

VoIP Server Parameters

Parameter	Permitted Values	Default
volpProt.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.		
volpProt.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.		
volpProt.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		
volpProt.server.x.address	**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.		
volpProt.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).		
volpProt.server.x.expires.lineSeize	**positive integer, minimum 10**	**30**
Requested line-seize subscription period.		
volpProt.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.		
volpProt.server.x.failOver.failBack.mode	**newRequests, DNSTTL, registration, duration**	**duration**
Specify the 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 `voIpProt.server.x.failOver.failBack.timeout`.		

VoIP Server Parameters (continued)

Parameter	Permitted Values	Default
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.

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.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.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.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**

VoIP Server Parameters (continued)

Parameter	Permitted Values	Default
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
volpProt.server.x.subscribe.expires	10 – 2147483647 seconds	3600 seconds

The phone's requested subscription period in seconds after which the phone attempts to resubscribe at the beginning of the overlap period. For example, if expires="300" and overlap="5", the phone resubscribes after 295 seconds (300–5). Note that the period negotiated with the server may be different.

Parameter	Permitted Values	Default
volpProt.server.x.subscribe.expires.overlap	5 – 65535 seconds	60 seconds

The number of seconds before the expiration time returned by server x after which the phone attempts to resubscribe. If the server value is less than the configured overlap value, the phone tries to resubscribe at half the expiration time returned by the server.

Parameter	Permitted Values	Default
volpProt.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.

Parameter	Permitted Values	Default
volpProt.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.

Parameter	Permitted Values	Default
volpProt.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).

Parameter	Permitted Values	Default
volpProt.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.

VoIP Server Parameters (continued)

Parameter	Permitted Values	Default
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**	**duration**

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.

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.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.

VoIP Server Parameters (continued)

Parameter	Permitted Values	Default
volpProt.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.		
volpProt.server.x.specialInterop	**VVX 101 = Standard, GENBAND, GENBAND-A2, ALU-CTS, DT** **VVX 201 = Standard, GENBAND, GENBAND-A2, ALU-CTS, ocs2007r2, lync2010** **All other phones = Standard, GENBAND, GENBAND-A2, ALU-CTS, DT, ocs2007r2, lync2010, lcs2005**	**Standard**
Enables server-specific features for all registrations.		
volpProt.server.x.subscribe.expires	**10 – 2147483647 seconds**	**3600 sec**
The phone's requested subscription period in seconds after which the phone attempts to resubscribe at the beginning of the overlap period. For example, if expires="300" and overlap="5", the phone resubscribes after 295 seconds (300–5). Note that the period negotiated with the server may be different.		
volpProt.server.x.subscribe.expires.overlap	**5 – 65535 seconds**	**60 seconds**
The number of seconds before the expiration time returned by server x after which the phone attempts to resubscribe. If the server value is less than the configured overlap value, the phone tries to resubscribe at half the expiration time returned by the server.		
volpProt.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.		
volpProt.server.H323.x.address	**IP address or hostname**	**Null**
Address of the H.323 gatekeeper. Note: Only one H.323 gatekeeper per phone is supported; if more than one is configured, only the first is used.		

VoIP Server Parameters (continued)

Parameter	Permitted Values	Default
voIpProt.server.H323.x.port	**0 to 65535**	**1719**
Port to be used for H.323 signaling. Note: The H.323 gatekeeper RAS signaling uses UDP, while the H.225/245 signaling uses TCP.		
voIpProt.server.H323.x.expires	**positive integer**	**3600**
Desired registration period.		

[1] Change causes phone to restart or reboot.

<SDP/>

The next table describes Session Description Protocol configuration parameters.

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. Note: If the H.323 call from a Polycom VVX 1500 selects a lower-quality codec (H.261) but the called device also support H.264, this parameter should be enabled to resolve the situation.		
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 the section <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.		

Session Description Protocol (SDP) Parameters (continued)

Parameter	Permitted Values	Default
voIpProt.SDP.offer.rtcpVideoCodecControl	**0 or 1**	**0**

This parameter determines whether or not RTCP-FB-based controls are offered in Session Description Protocol (SDP) when the phone negotiates video I-frame request methods. Even when RTCP-FB-based controls are not offered in SDP, the phone may still send and receive RTCP-FB I-frame requests during calls depending on other parameter settings. For more information about video I-frame request behavior, refer to video.forceRtcpVideoCodecControl. For an account of all parameter dependencies refer to the section Configure I-Frames.

If 1, the phone adds the SDP attribute "a=rtcp-fb" into offers during outbound SIP calls. If 0, the phone does not include the SDP attribute "a=rtcp-fb".

<SIP/>

The next table describes SIP configuration parameters.

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.

Parameter	Permitted Values	Default
voIpProt.SIP.alertInfo.x.value	**string**	**Null**

A string to match the Alert-Info header in the incoming INVITE.

Parameter	Permitted Values	Default
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.

Parameter	Permitted Values	Default
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.

Session Initiation Protocol (SIP) Parameters (continued)

Parameter	Permitted Values	Default
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.		
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 index) 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`.		

Session Initiation Protocol (SIP) Parameters (continued)

Parameter	Permitted Values	Default
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**
Enable or disable DTMF relays for active SIP calls. Not supported for H.323 calls. 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.dtmfViaSignaling.rfc2976.nonLegacyEncoding[1]	**0 or 1**	**0**
Controls the behavior of the Star and Pound keys used for DTMF relays for active SIP calls. Not supported for H.323 calls. If set to 1, the phone sends an asterisk (*) when the Star key is pressed and a hashtag (#) when the Pound key is pressed. If set to 0, the phone sends 10 when the Star key (*) is pressed and 11 when the Pound key (#) is pressed.		
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. You must use a registration expiry of 66 seconds or greater for failover with a 503 response to work properly. This rule applies both to the phone configuration (`reg.x.server.y.expires` and `voIpProt.server.x.expires`) as well as the 200 OK register response from the server.		
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.		
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<XREF>.		
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.		

Session Initiation Protocol (SIP) Parameters (continued)

Parameter	Permitted Values	Default
voIpProt.SIP.intercom.alertInfo	**Alpha-Numeric string**	**Intercom**

The string you want to use in the Alert-Info header. You can use the following characters: '@', '-' ,'_' , '.' .
If you use any other characters, NULL, or empty spaces, the call is sent as normal without the Alert-Info header.

Parameter	Permitted Values	Default
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.

Parameter	Permitted Values	Default
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).

Parameter	Permitted Values	Default
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.

Parameter	Permitted Values	Default
voIpProt.SIP.looseContact	**0 or 1**	**0**

0 (default) - The ephemaral port is added to the contact header in TLS case.
1 - The port parameter is not added to the contact header or SIP messages.

Parameter	Permitted Values	Default
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.

Parameter	Permitted Values	Default
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.newCallOnUnRegister	**0 or 1**	**1**

If set to 0 , the phone does not generate new Call-ID and From tag during re-registration.

Parameter	Permitted Values	Default
voIpProt.SIP.outboundProxy.address	**IP address or hostname**	**Null**

The IP address or hostname of the SIP server to which the phone sends all requests.

Parameter	Permitted Values	Default
voIpProt.SIP.outboundProxy.port	**0 to 65535**	**0**

The port of the SIP server to which the phone sends all requests.

Session Initiation Protocol (SIP) Parameters (continued)

Parameter	Permitted Values	Default
volpProt.SIP.outboundProxy.failOver.failBack.mode	**newRequests, DNSTTL, registration, duration,**	**duration**

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.

volpProt.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.

volpProt.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.

volpProt.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`.

volpProt.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.

volpProt.SIP.outboundProxy.transport	**DNSnaptr, TCPpreferred, UDPOnly, TLS, TCPOnly**	**DNSnaptr**

Session Initiation Protocol (SIP) Parameters (continued)

Parameter	Permitted Values	Default
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.		
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.		

Session Initiation Protocol (SIP) Parameters (continued)

Parameter	Permitted Values	Default
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.callRecording	**0 or 1**	**0**
Enable or disable the BroadSoft BroadWorks v20 call recording feature for multiple phones.		
voIpProt.SIP.serverFeatureControl.cf[1]	**0 or 1**	**0**
If set to 1, server-based call forwarding is enabled. Server and local phone call-forwarding are synchronized. If set to 0, server-based call forwarding is not enabled. Requires server-side support of synchronized call forwarding.		
voIpProt.SIP.serverFeatureControl.dnd[1]	**0 or 1**	**0**
If set to 1, server-based DND is enabled. Server and local phone DND are synchronized. If set to 0, server-based DND is not enabled. Requires server-side support of synchronized do not disturb (DND).		
voIpProt.SIP.serverFeatureControl.localProcessing.cf	**0 or 1**	**1**
This parameter depends on the value of `voIpProt.SIP.serverFeatureControl.cf`. If set to 0 and `voIpProt.SIP.serverFeatureControl.cf` is set to 1, call forwarding is performed on the server side only, and the phone does not perform local call forwarding. If set to 1 and `voIpProt.SIP.serverFeatureControl.cf` is set to 1, the phone and the server perform call forwarding. If both `voIpProt.SIP.serverFeatureControl.localProcessing.cf` and `voIpProt.SIP.serverFeatureControl.cf` are set to 0, the phone performs local call forwarding and the `localProcessing` parameter is not used.		
voIpProt.SIP.serverFeatureControl.localProcessing.dnd	**0 or 1**	**1**
This parameter depends on the value of `voIpProt.SIP.serverFeatureControl.dnd`. If set to 0 and `voIpProt.SIP.serverFeatureControl.dnd` is set to 1, do not disturb (DND) is performed on the server-side only, and the phone does not perform local DND. If set to 1 and `voIpProt.SIP.serverFeatureControl.dnd` is set to 1, the phone and the server perform DND. If both `voIpProt.SIP.serverFeatureControl.localProcessing.dnd` and `voIpProt.SIP.serverFeatureControl.dnd` are set to 0, the phone performs local DND and the `localProcessing` parameter is not used.		
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.securityClassification	**0 or 1**	**0**
Enable or disable the visual security classification feature for all lines on a phone.		
voIpProt.SIP.specialEvent.checkSync.alwaysReboot	**0 or 1**	**0**

Session Initiation Protocol (SIP) Parameters (continued)

Parameter	Permitted Values	Default
If set to 1, always reboot when a NOTIFY message is received from the server with event equal to check-sync even if there has not been a change to software or configuration. If set to 0, the phone will only reboot if necessary. Many configuration parameter changes can be applied dynamically without the need for a reboot.		
voIpProt.SIP.specialEvent.checkSync.downloadCallList	**0 or 1**	**0**
If 1, the phone downloads the call list for the logged-in user when a check sync event's NOTIFY message is received from the server. If 0, the phone does not download the call list for the logged-in user when a check sync event's NOTIFY message is received from the server.		
voIpProt.SIP.specialEvent.checkSync.downloadDirectory	**0 or 1**	**0**
If set to 0, the phone only downloads software and configuration updates after receiving a `checksync NOTIFY` message. If set to 1, the phone downloads the updated global and personal directory files along with any software and configuration updates after receiving a `checksync NOTIFY` message. The files are downloaded when the phone restarts, reboots, or when the phone downloads any software or configuration updates.		
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.		
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.strictReplacesHeader	**0 or 1**	**1**
This parameter applies only to directed call pick-up attempts initiated against monitored BLF resources. If set to 1, the phone requires call-id, to-tag, and from-tag to perform a directed call-pickup when `call.directedCallPickupMethod` is configured as `native`. If set to 0, call pick-up requires a call id only.		
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**

Session Initiation Protocol (SIP) Parameters (continued)

Parameter	Permitted Values	Default
If 0, TLS DSK is disabled. If 1, TLS DSK is enabled. For more information, see Protocol Overview on Microsoft Developer Network.		
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.		
voipPort.SIP.useCompleteUriForRetrieve	**0 or 1**	**1**
If set to 1, the target URI in BLF signaling will use the complete address as provided in the xml dialog document. If set to 0, only the user portion of the XML dialog document is used and the current registrar's domain is appended to create the full target URI.		
voipPort.SIP.useLocalTargetUriforLegacyPickup	**0 or 1**	**1**
If set to 1, BLF signaling will use the address as provided in the local target URI in xml dialog document with additional rules based on `voipPort.SIP.useCompleteUriForRetrieve`. If set to 0, the local target uri is not considered and the identity attribute is used with additional rules based on `voipPort.SIP.useCompleteUriForRetrieve`.		
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.useRFC3264HoldOnly	**0 or 1**	**0**
If set to 1, and no media direction is specified, the phone uses `sendrecv` compliant with RFC 3264 when negotiating SDP and generates responses containing RFC 3264-compliant media attributes for calls placed on and off hold by either end. If set to 0, and no media direction is specified, the phone enters backward compatibility mode when negotiating SDP and responds using the c=0.0.0.0 RFC 2543 signaling method. Note: `voIpProt.SIP.useSendonlyHold` applies only to calls on phones that originate the hold.		
voIpProt.SIP.useSendonlyHold	**0 or 1**	**1**

Session Initiation Protocol (SIP) Parameters (continued)

Parameter	Permitted Values	Default

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).

[1] Change causes phone to restart or reboot.

<H323/>

The parameters listed in the next table are supported only with the Polycom VVX 500/501, 600/601, and 1500 phones.

H.323 Protocol Parameters

Parameter	Permitted Values	Default
voIpProt.H323.autoGateKeeperDiscovery[1]	**0 or 1**	**1**

If set to 1, the phone will attempt to discover an H.323 gatekeeper address via the standard multicast technique, provided that a statically configured gatekeeper address is not available.

If set to 0, the phone will not send out any gatekeeper discovery messages.

voIpProt.H323.blockFacilityOnStartH245[1]	**0 or 1**	**0**

If set to 1, facility messages when using H.245 are removed.

voIpProt.H323.dtmfViaSignaling.enabled[1]	**0 or 1**	**1**

If set to 1, the phone will use the H.323 signaling channel for DTMF key press transmission.

voIpProt.H323.dtmfViaSignaling.H245alphanumericMode[1]	**0 or 1**	**1**

If set to 1, the phone will support H.245 signaling channel alphanumeric mode DTMF transmission.

Note: If both alphanumeric and signal modes can be used, the phone gives priority to DTMF.

voIpProt.H323.dtmfViaSignaling.H245signalMode[1]	**0 or 1**	**1**

If set to 1, the phone will support H.245 signaling channel signal mode DTMF transmission.

voIpProt.H323.enable[1]	**0 or 1**	**0**

A flag to determine if the H.323 protocol is used for call routing, dial plan, DTMF, and URL dialing.

If set to 1, the H.323 protocol is used.

voIpProt.H323.local.port[1]	**0 to 65535**	**1720**

Local port to be used for H.323 signaling. Local port for sending and receiving H.323 signaling packets.

If set to 0, 1720 is used for the local port but is not advertised in the H.323 signaling.

If set to some other value, that value is used for the local port and it is advertised in the H.323 signaling.

voIpProt.H323.local.RAS.port[1]	**1 to 65535**	**1719**

H.323 Protocol Parameters (continued)

Local port for RAS signaling.

[1] Change causes phone to restart or reboot.

<VVXD60/>

The parameters listed in the following table set the line and line termination for VVX D60 Wireless Handsets registered to VVX business media phones.

VVX D60 Accessory Parameters

Parameter	Permitted Values	Default
VVXD60.Handset.X.outGoingLineIndex	**1-34**	**1**

Controls the registration index that is used as the default line for outgoing calls placed on the wireless handset without selecting a line first. X refers to the wireless handset where x can be 1-5.

VVXD60.Handset.X.line.Y	**0-34**	**0**

Sets the lines that will be accessible from the wireless handset where X is the wireless handset (1-5) and Y is the registered line on the VVX phone that will be mapped to the wireless handset. You can map up to five lines to a wireless handset.

For example, you can configure the 4 lines on a VVX phone to the two registered wireless handsets.

Existing VVX configuration:

- reg.1.address = "1000"
- reg.2.address = "2000"
- reg.3.address = "3000"
- reg.4.address = "4000"

To configure Handset 1 with the line 2000 and Handset 3 with lines 2000 and 4000, use the following configuration:

- VVXD60.Handset.1.line.1 = "2"
- VVXD60.Handset.2.line.1 = "2"
- VVXD60.Handset.2.line.2 = "4"

The parameters listed in the next table specify the download location of the translated language files for the Web Configuration Utility.

Web Configuration Utility Parameters

Parameter	Permitted Values	Default
webutility.language.plcmServerUrl	**URL**	`http://downloads.polycom.com/voice/software/languages/`

The download location of the translated language files for the Web Configuration Utility.

The parameters in the following table set the XML streaming protocols for instant messaging, presence, and contact list for BroadSoft features.

XML Streaming Protocol Parameters

Parameter	Permitted Values	Default
xmpp.1.auth.domain	**UTF-8 encoded string**	**Null**
Specify the domain name of the XMPP server.		
xmpp.1.auth.password	**UTF-8 encoded string**	**Null**
Password used for XMPP registration. Specify the password for XMPP registration.		
xmpp.1.auth.useLoginCredentials	**0 or 1**	**0**
Choose whether or not to use the login credentials provided in the phone's Login Credentials Menu for XMPP authentication.		
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**
Enable or disable XMPP presence.		
xmpp.1.jid	**String min 0, max 256**	**Null**
Enter the Jabber identity used to register with the presence server, for example: `presence.test2@polycom-alpha.eu.bc.im`.		
xmpp.1.roster.invite.accept	**Automatic or prompt**	**prompt**
Choose how phone users receive the BroadSoft XMPP invitation to be added to a buddy list. If set to prompt, the phone displays a list of users who have requested to add you as a buddy and you 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 an IP address, host name, or FQDN, for example: `polycom-alpha.eu.bc.im`.		
xmpp.1.verifyCert	**0 or 1**	**1**
Enable or disable verification of the TLS certificate provided by the BroadSoft XMPP presence server.		

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 information on:

- **Basic Protocols**

  All 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.

RFC and Internet Draft Support

The following RFCs and Internet drafts are supported. For more information on any of the documents, go to Request for Comments (RFC) and enter the RFC number in the search box.

Supported RFC and Internet Drafts

RFC or Draft	Notes
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. All sections RFC 3265 supported with the exception of Section 3.3.3 Forking and Section 4.4.9 Handling of forked requests.
RFC 3311	The Session Initiation Protocol (SIP) UPDATE Method

Supported RFC and Internet Drafts

RFC or Draft	Notes
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 3581	An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing
RFC 3611	RTP Control Protocol Extended reports (RTCP XR)
RFC 3608	Session Initiation Protocol (SIP) Extension Header Field for Service Route Discovery During Registration
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 3680	A Session Initiation Protocol (SIP) Event Package for Registrations
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)
RFC 5009	Private Header (P-Header) Extension to the Session Initiation Protocol (SIP) for Authorization of Early Media
RFC 6026	Correct Transaction Handling for 2xx Responses to Session Initiation Protocol (SIP) INVITE Requests

Supported RFC and Internet Drafts

RFC or Draft	Notes
RFC 6228	Session Initiation Protocol (SIP) Response Code for Indication of Terminated Dialog
RFC 6665	SIP-Specific Event Notification
RFC 6947	The Session Description Protocol (SDP) Alternate Connectivity (ALTC) Attribute
RFC 7329	Session Identifier for the Session Initiation Protocol (SIP)
RFC 7462	URNs for the Alert-Info Header Field of 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 listed in the following table are supported.

Supported SIP Request Messages

Method	Supported	Notes
REGISTER	Yes	
INVITE	Yes	
ACK	Yes	
CANCEL	Yes	
BYE	Yes	
OPTIONS	Yes	
SUBSCRIBE	Yes	
NOTIFY	Yes	
REFER	Yes	

Supported SIP Request Messages

Method	Supported	Notes
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	

Supported SIP Request Headers

The following table lists the SIP request headers and indicates which are supported. Headers with 'Yes' in the Supported column indicates that 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
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

Supported SIP Request Headers

Header	Supported
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
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

Supported SIP Request Headers

Header	Supported
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
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

Supported SIP Request Headers

Header	Supported
Warning	Yes (Only warning codes 300 to 399)
WWW-Authenticate	Yes
X-Sipx-Authidentity	Yes

Response Support

The following tables list the SIP responses and indicates which are supported. Responses with 'Yes' in the Supported column indicates that the header is sent and properly parsed. The phone might not generate the response.

- Supported 1xx SIP Responses
- Supported 2xx SIP Responses
- Supported 3xx SIP Responses
- Supported 4xx SIP Responses
- Supported 5xx SIP Responses
- Supported 6xx SIP Responses

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
302 Moved Temporarily	Yes
305 Use Proxy	No
380 Alternative Service	No

4xx Responses - Request Failure

All 4xx responses for which the phone does not provide specific support will be treated the same as 400 Bad Requests.

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
421 Extension Required	No
423 Interval Too Brief	Yes

Supported 4xx SIP Responses

Response	Supported
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

Supported 6xx SIP Responses

604 Does Not Exist Anywhere	No
606 Not Acceptable	No

Hold Implementation

The phone supports two currently accepted means of signaling hold. The phone can be configured to use either hold signaling method (refer to <SIP/> parameters). The phone supports both methods when signaled by the remote endpoint.

Supported Hold Methods

Method	Notes
Signal the media directions with the "a" SDP media attributes sendonly, recvonly, inactive, or sendrecv.	Preferred method.
Set the "c" destination addresses for the zmedia streams in the SDP to zero. For example, c=0.0.0.0	No longer recommended due to RTCP problems associated with this method. Receiving sendrecv, sendonly, or inactive from the server causes the phone 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 can be used for remote control of the phone from computer applications.

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 <SIP/> and <reg/> 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 Skype for Business. 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 (SCA) 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.

Polycom devices support Shared Call Appearance (SCA) using the SUBSCRIBE-NOTIFY method specified in RFC 6665. 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 event used is "dialog" for bridged line appearance subscribe and notify.

Monitoring, Diagnostics, and Troubleshooting

Polycom phones running Polycom UC Software provide a variety of screens and logs that allow you to review information about the phone and its performance, help you diagnose and troubleshoot problems, view error messages, and test the phone's hardware.

Review the latest UC Software Release Notes on Polycom UC Software Support Center for known problems and possible workarounds. If you don't find your problem in this section or in the latest Release Notes, contact your Certified Polycom Reseller for support.

Error Message Types

The following sections cover some of the errors you might see, along with suggested actions.

Updater Error Messages

If a fatal error occurs, the phone does not boot up. If the error is not fatal, the phone boots up but its configuration might be changed. Most updater errors are logged to the phone's boot log. However, if the phone is having trouble connecting to the provisioning server, the phone is not likely to upload the boot log.

The following table describes possible solutions to updater error messages.

Updater Error Messages

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.
Application <file name> is not compatible with this phone!
An application file was downloaded from the provisioning server, but it cannot be installed on this phone.   Install a compatible software image on the provisioning server. Be aware that there are various hardware and software dependencies.
Could not contact boot server using existing configuration
The phone cannot contact the provisioning server. Possible causes include: - Cabling issues   - DHCP configuration   - Provisioning server problems   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.

Updater Error Messages

Error, application is not present!

The phone does not have an application stored in device settings and, because the application could not be downloaded, the phone cannot boot.

- Download compatible Polycom UC Software to the phone using one of the supported provisioning protocols.

Polycom UC Software Error Messages

If an error occurs in the UC Software, phones running UC Software 4.0.0 or later display an error message and a warning icon in the phone's interface. Depending on the phone model, users can navigate to the list of warnings using the menu paths indicated:

- ➢ VVX 1500   **Menu > Status > Diagnostics > Warnings**
- ➢ VVX 300/301, 310/311, 400/401, 410/411, 500/501, or 600/601   **Settings > Status > Diagnostics > Warnings**

The following table describes Polycom UC Software error messages.

Polycom UC Software Error Messages

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 the configuration files contain these deprecated parameters:
- 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 displays if any configuration file contains more than 100 of the following errors:
- Unknown parameters
- Out-of-range values
- Invalid values.

To check that your configuration files use correct parameter values, refer to Using Correct Parameter XML Schema, Value Ranges, and Special Characters.

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 on the phone: **Status** > **Basic** > **Login Credentials**.

Missing files, config. reverted

Polycom UC Software Error Messages

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 also displays if the files listed in the **<MAC Address>.cfg** file are not present on the provisioning server.

Network link is down

Indicates that the phone cannot establish a link to the network and persists until the link problem is resolved. Call-related functions, soft keys, and line keys are disabled when the network is down but the phone menu works.

Network Authentication Failure Error Codes

This message displays if 802.1X authentication with the Polycom phone fails. The error codes display on the phone when you press the **Details** key. Error codes are also included in the log files.

Network Authentication Failure Error Codes

Event Code	Description	Comments
1	Unknown events	An unknown event by '1' can include any issues 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 'xxx' is the standard TLS alert message code. For example, if the phone presents a certificate with invalid signature and/or content, 'xxx' is 042. For the generic certificate error code, 'xxx' is 000.	See section 7.2 of RFC 2246 for further TLS alert codes and error codes.
31xxx	Server Certificate failure 'xxx' can use 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 'xxx' is the TLS alert message code). For example, if the protocol version presented by the server is not supported by the phone, then 'xxx' is 70, and the EAP error code is 4070.	See section 7.2 of RFC 2246 for further TLS alert codes and error codes.

Status and Diagnostics

The phone includes a variety of information screens and tools that can help you monitor the phone and resolve problems.

View the Phone's Status

You can troubleshoot phone issues by viewing the phone's Status menu.

To view the Status menu on the phone:

   1   Select **Menu** > **Status** > **Select**.

   2   Scroll to a Status menu item and press **Select**. The following table lists available options:

Status Menu Descriptions

Menu Item	Menu Information
Platform	• Phone's serial number or MAC address • Current IP address • Updater version • Application version • Name of the configuration files in use • Address of the provisioning server
Network	• TCP/IP Setting • Ethernet port speed • Connectivity status of the PC port (if it exists) • Statistics on packets sent and received since last boot • Last time the phone rebooted • Call Statistics showing packets sent and received on the last call
Lines	• Detailed status of each of the phone's configured lines
Diagnostics	• Hardware tests to verify correct operation of the microphone, speaker, handset, and third party headset, if present • Tests to verify proper functioning of the phone keys • List of the functions assigned to each of the phone keys • Real-time graphs for CPU, network, and memory use

Test Phone Hardware

You can test the phone's hardware directly from the user interface.

To test phone hardware:

   1   Go to **Menu** > **Settings** > **Status** > **Diagnostics.**

   2   Choose from these tests:

   ➢ **Audio Diagnostics**   Test the speaker, microphone, handset, and a third party headset**.**

   ➢ **Keypad Diagnostics**   Verify the function assigned to each keypad key.

   ➢ **Display Diagnostics**   Test the LCD for faulty pixels.

   ➢ **LED Diagnostics**   Test the LED lights on your phone.

   ➢ **Touch Screen Diagnostics**   Test the touch screen response.

Upload a Phone's Configuration

If your phone is running Polycom UC Software 3.3.0 or later, you can upload the phone's current configuration files to help you debug configuration problems. 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.

If your phone is running Polycom UC Software 4.0.0 or later, you can use the Web Configuration Utility to upload the files.

To upload the phone's current configuration:

1   Navigate to **Menu** > **Settings** > **Advanced** > **Admin Settings** > **Upload Configuration**.

2   Choose which files to upload: **All Sources**, **Configuration Files**, or **Web**. If you use the Web Configuration Utility, you can also upload **Device Settings**.

3   Press **Upload**.

4   The phone uploads the configuration file to the location you specified in the parameter `prov.configUploadPath`.

   For example, if you select **All Sources**, a file ***<MACaddress>*-update-all.cfg** is uploaded.

Perform Network Diagnostics

If your phone is running UC Software 4.0.0 or later, you can use ping and traceroute to troubleshoot network connectivity problems.

To use network diagnostics tools:

1   Go to **Menu > Status > Diagnostics > Network.**

2   Enter a URL or IP address.

3   Press **Enter**.

Log File Format

You can configure Polycom phone logging to suit your needs. Log file names use the following format:

`MAC address]_[Type of log].log`

For example, if the MAC address of your phone is **0004f2203b0**, the app log file name is **0004f2203b0-app.log**.

Configure Severity of Events Logged

You can configure the severity of the events that are logged independently for each module of the Polycom UC Software. This enables you to capture lower severity events in one part of the application, and high severity events for other components. Severity levels range from 0 to 6, where 0 is the most detailed logging and 6 captures only critical errors. Note that user passwords display in level 1 log files.

You must contact Polycom Customer Support to obtain the template file `techsupport.cfg` containing parameters that configure log levels.

Severity of Events Logged

Parameter Template	Permitted Values
`log.level.change.module_name` techsupport.cfg	Specifies the severity level logged for the specified module. Not all modules are available for all phone models.

Configure Log File Collection and Storage

You can configure log file collection and storage using the parameters in the following table.

You must contact Polycom Customer Support to obtain the template file `techsupport.cfg` containing parameters that configure log file collection and storage.

Log File Collection and Storage Parameters

Parameter Template	Permitted Values
`log.render.level` techsupport.cfg	Sets the lowest level that can be logged. 1 (default)
`log.render.file.size` techsupport.cfg	Sets the maximum file size in kilobytes before the log is uploaded 32 kb (default)
`log.render.file.upload.period` techsupport.cfg	Number of seconds between log uploads 172800 (default) - 48 hours
`log.render.file.upload.append` techsupport.cfg	Specify whether uploaded log files overwrite existing files or are appended to existing files. 1 (default) Note that this parameter is not supported by all servers.
`log.render.file.upload.append.sizeLimit` techsupport.cfg	Specify the maximum size in kilobytes of log files on the provisioning server. 512kb (default)
`log.render.file.upload.append.limitMode` techsupport.cfg	Specify whether to stop or delete logging when the server log reaches its maximum size. delete (default) stop -

Use Scheduled Logging

Scheduled logging can help you monitor and troubleshoot phone issues. Use the parameters in this table to configure scheduled logging.

You must contact Polycom Customer Support to obtain the template file `techsupport.cfg` containing parameters that configure scheduled logging.

Scheduled Logging Parameters

Parameter Template	Permitted Values
`log.sched.module_name` techsupport.cfg	

Upload Logs Manually

You can manually initiate a log upload by pressing the correct multiple key combination on the phone (see Define the Phone Key Layout).

When you manually upload log files, the word *now* is inserted into the name of the file, for example, **0004f200360b-now-boot.log**.

Read Log Files

The phone writes information into several different log files. This table describes the type of information in each.

Log File Descriptions

Log File	Description
Boot Log	
Application Log	
Syslog	For more information about Syslog, see Syslog on Polycom Phones - Technical Bulletin 17124.

Monitoring the Phone's Memory Usage

To ensure that your phones and their configured features operate smoothly, verify that the phones have adequate available memory resources. If you are using a range of phone features, customized configurations, or advanced features, you might need to manage phone memory resources.

If your deployment includes a combination of phone models, consider configuring each phone model separately with its own features instead of applying all phone features to all phone models.

For best performance, the phone should use no more 95% of its available memory. When the phone memory resources are 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 running or do not start.

Check Memory Usage from the Phone

You can view a graphical representation of the phone's memory usage directly on the phone.

1 Load and configure the features and files you want to make available on the phone's interface.

2 Navigate to **Status > Diagnostics > Graphs > Memory Usage.**

View Memory Usage Errors in the Application Log

Each time the phone's minimum free memory goes below about 5%, the phone posts a "Minimum free memory reached" error message in the application log.

The application log file is enabled by default. The file is uploaded to the provisioning server directory on a schedule you configure. You can also upload a log file manually. For information on manually uploading log files, refer to Upload Logs Manually.

Manage Phone Memory Resources

If you need to free memory on your phone, review the following table for the amount of memory each customizable feature uses and consider strategies for reducing the amount of memory you need the feature to use.

Managing Phone Memory Resources

Feature	Typical Memory Size
Idle Browser	**Varies, depending on number and complexity of application elements.**
To reduce memory resources used by the idle browser: • Display no more than three or four application elements. • Simplify pages that include large tables or images.	
Custom Idle Display Image	**15 KB**
The average size of the Polycom display image is 15 KB. Custom idle display image files should also be no more than 15 KB.	
Main Browser	**Varies, depending on number and complexity of applications.**
To reduce memory resources used by the main browser: • Display no more than three or four application elements. • Simplify pages.	
Local Contact Directory	**42.5 KB**

Managing Phone Memory Resources

Feature	Typical Memory Size
Polycom phones are optimized to display a maximum of 250 contacts. Each contact has four attributes and requires 170 bytes. A local contact directory of this size requires 42.5 KB. To reduce memory resources used by the local contact directory: • Reduce the number of contacts in the directory • Reduce the number of attributes per contact	
Corporate Directory	**Varies by server**
Polycom phones are optimized to corporate directory entries with 5 - 8 contact attributes each. The size of each entry and the number of entries in the corporate directory vary by server. If the phone is unable to display directory search results with more than five attributes, make additional memory resources available by reducing memory requirements of another feature.	
Ringtones	**16 KB**
The Polycom ringtone files range in size from 30KB to 125KB. If you use custom ringtones, Polycom recommends limiting the file size to 16KB. To reduce memory resources required for ringtones: • Reduce the number of available ringtones.	
Background Images	**8 – 32 KB**
Polycom phones are optimized to display background images of 50KB. To reduce memory resources required for background images: • Reduce the number and size of available background images.	
Phone Interface Language	**90 - 115 KB, depending on language**
The language dictionary file used for the phone's user interface ranges from 90KB to 115KB for languages that use an expanded character set. To conserve memory resources, Polycom recommends using XML language files for only the languages you need.	
Web Configuration Utility Interface	**250 KB - 370 KB**
The language dictionary file used for the Web Configuration Utility interface ranges from 250KB to 370KB for languages that use an expanded character set. To conserve memory resources, Polycom recommends using XML language files for only the languages you need.	

Troubleshooting

This section lists potential issues, problems, and common difficulties and possible solutions.

Power and Startup Issues

The following table describes possible solutions to power and startup issues.

Troubleshooting Power and Startup Issues

The phone has power issues or the phone has no power.

Determine whether 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 to see if the phone is properly plugged into a functional AC outlet.
* Make sure that the phone is not plugged into an outlet controlled by a light switch that is turned off.
* If the phone is plugged into a power strip, try plugging directly into a wall outlet instead.

The phone does not boot.

If the phone does 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 configured with the correct address for the provisioning server on the network.

Dial Pad Issues

The following table describes possible solutions to issues with the dial pad.

Troubleshooting Dial Pad Issues

The dial pad does not work.

If the dial pad on your phone does not respond, do one of the following:
* Check for a response from other feature keys.
* Place a call to the phone from a known working telephone. Check for display updates.
* On the phone, go to **Menu > System Status > Server Status** to check if the telephone is correctly registered to the server.
* On the phone, go to **Menu > System Status > Network Statistics**. Scroll down to see whether LAN port shows Active or Inactive.
* Check the termination at the switch or hub end of the network LAN cable. Ensure that the switch/hub port that is connected to the telephone is operational.

Screen and System Access Issues

The following table 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 keys do not respond to presses:
- 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.
- On the phone, go to Navigate to **Menu > Status > Lines** to confirm the line is actively registered to the call server.
- Reboot the phone to attempt re-registration to the call server.

The display shows the message *Network Link is Down*.

This message displays when the LAN cable is not properly connected. Do one of the following:
- Check the termination at the switch or hub end of the network LAN cable.
- Check that the switch or hub is operational (flashing link/status lights).
- On the phone, go to **Menu > Status > Network**. Scroll down to verify that the LAN is active.
- Ping the phone from a computer.
- Reboot the phone to attempt re-registration to the call server. Navigate **to Menu > Settings > Advanced > Reboot Phone**).

Calling Issues

The following table provides possible solutions to 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 you are using in-line powering, check that the switch is supplying power to the phone.

The dial tone is not present on one of the audio paths.

If dial tone is not present on one of the audio paths, do one of the following:
- Switch between handset, headset (if present), or handsfree speakerphone to see whether or not dial tone is present on another path.
- If the dial tone exists on another path, connect a different handset or headset to isolate the problem.
- Check configuration for gain levels.

The phone does not ring.

If there is no ringtone 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.
- Check the status of handset, headset (if connected), and handsfree speakerphone.

Troubleshooting Calling Issues

The line icon shows an unregistered line icon.

If the phone displays an icon indicating that a line is unregistered, do the following:
- Try to re-register the line and place a call.

Display Issues

The following table 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 cable is inserted properly at the rear of the phone.
- If your are using PoE powering, check that the PoE switch is supplying power to the phone.
- Use the screen capture feature to verify whether the screen displays properly in the capture. Refer to Capture Your Device'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.
- Reboot the phone to obtain the default level of contrast.
- Use the screen capture feature to verify whether the screen displays properly in the capture. Refer to Capture Your Device'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:
- Angle or move the Polycom phone away from the lights.

The time and date are flashing.

If the time and date are flashing, the phone is disconnected from the LAN or there is no SNTP time server configured. Do one of the following:
- Reconnect the phone to the LAN.
- Configure an SNTP server.
- Disable the time and date if you do not want to connect your phone to a LAN or SNTP server.

Audio Issues

The following table describes possible solutions to audio issues.

Troubleshooting Audio Issues

There is no audio on the headset

If there is no audio on your headset, the connections may not be correct. Do one of the following:
- Ensure the headset is plugged into the jack marked Headset at the rear of the phone.
- Ensure the headset amplifier (if present) is turned on and adjust the volume.

Licensed Feature Issues

The following table describes issues for features that require a license.

Troubleshoot Feature License Issues

Voice Quality Monitoring or H.323 is not available on the phone.

If you cannot access features, check your licenses on the phone by navigating to **Menu > Status > Licenses**.
- You require a license key to activate the VQMon feature on the following VVX business media phones: 101, 201, 300, 301, 310, 311, 400, 401, 410, 411. The following phones do not require a license key to activate the VQMon feature: 500, 600, 1500, Polycom RealPresence Trio.
- You need a license to use H.323 on VVX 1500. You do not need a license to use H.323 on the VVX 500/501, 600/601. Note that H.323 is not supported on VVX 300/301, 310/311, 400/401, 410/411, and SoundStructure VOIP Interface.
- If your phone is not installed with UC Software version 4.0.0 or later, you also require a license for conference management, corporate directory, and call recording.

Software Upgrade Issues

The following table describes possible solutions to issues that may occur during or after a software upgrade.

Troubleshooting Software Upgrade Issues

Some 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 **Menu > Status > Platform > Configuration**. If there are messages stating *Errors Found*, *Unknown Params,* or *Invalid values*, correct your configuration files and restart the phone.

The phone displays a *Config file error* message for five seconds after it boots up.

You are using configuration files from a UC Software version earlier than the UC Software image running on the phones. Configuration parameters and values can change each release and specific parameters may or may not be included.
- Correct the configuration files, remove the invalid parameters, and restart the phone.
- See the UC Software Administrator's Guide and Release Notes for the UC Software version you have installed on the phones.

Troubleshooting Software Upgrade Issues

When using the Web Configuration Utility to upgrade phone software, the phone is unable to connect to the Polycom Hosted Server.

Occasionally, the phone is unable to connect to the Polycom hosted server because of the following:

- The Polycom hosted server is temporarily unavailable.
- There is no 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 using the *Polycom UC Software Release Matrix for VVX Phones*.
- Verify that your network's configuration allows the phone to connect to http://downloads.polycom.com.
- If the issue persists, try manually upgrading your phone's software.

Wireless Handset Software Upgrade Issues

If any wireless handset fails to update its software, the base station retries pushing the update to the wireless handset three times before moving to the next registered handset. If the base station or wireless handset fails to update or restarts during the update process, the base station or wireless handset restarts with the previous software version.

Try the following solutions if the base station or any of the wireless handsets fail to update:

- Manually update the wireless handset software (refer to section Update the Wireless Handset Software Manually).
- Restart the base station, then pair the base station with the VVX business media phone again.
- To restart the base station, using a paper clip, press and hold the Reset button on the back of the base station for five seconds.
- After the software process is complete for all registered wireless handsets, unregister and register any wireless handsets that failed to update.

If the base station fails to pair with the VVX business media phone after successfully updating, you need to repair the base station with the phone manually (refer to section Pair the VVX D60 Base Station).

Ports Used on Polycom Phones

The following table lists the ports currently used by the Polycom UC Software.

Ports Used by Polycom Phones

Port Number	Protocol	Outgoing	Incoming	UDP or TCP
21	FTP	Provisioning, Logs		TCP
22	SSH	Admin	Admin	TCP
23	Telnet[1]	Admin		TCP
53	DNS			UDP

Ports Used by Polycom Phones

Port Number	Protocol	Outgoing	Incoming	UDP or TCP
67	DHCP	Server		UDP
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			
1719	H.323[2]	RAS Signaling	RAS Signaling	
1720	H.323[2]	Signaling	Signaling	
2222	RTP[3]	Media Packets	Media Packets	
2223	RTCP[3]	Media Packet Statistics	Media Packet Statistics	
5060	SIP	SIP signaling	SIP signaling	
5061	SIP over TLS	Secure signaling	Secure signaling	
24800	PDC	PDC Client messages	PDC Server messages	TCP

[1] Telnet is disabled by default on VVX phones.

[2] H.323 is available only on the VVX 500/501, 600/601, and 1500.

[3] RTP and RTCP can use any even port between 2222 and 2269 (2317 on VVX 500/501, 600/601, or 1500), but you can configure ports by setting `tcpIpApp.port.rtp.mediaPortRangeStart`.

Update and Maintain Polycom Devices and UC Software

This section provides information on updating and maintaining your devices and the UC Software.

You can upgrade the software that is running on the Polycom phones in your organization. The upgrade process varies with the version of Polycom UC Software that is currently running on your phones and with the version that you want to upgrade to.

- As of UC Software 5.3.0, you can update software with the user-controlled software update feature explained in the section User-Controlled Software Update.
- If you are updating software from UC Software 4.0.x, see the section Update Phones from UC Software 4.0.x.

The Updater, UC Software executable, and configuration files can all be updated using centralized provisioning.

Update 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 instructions, see *Use the Software Upgrade Tool in the Web Configuration Utility: Feature Profile 67993* at Polycom Engineering Advisories and Technical Notifications.

Configuration changes made to individual phones using the Web Configuration Utility override configuration settings made using central provisioning. For information about using multiple provisioning methods, refer to Polycom Provisioning Methods.

User-Controlled Software Update

This feature is available on VVX business media phones as of UC Software 5.3.0 and enables phone users to choose when to accept software updates the administrator sends to the phones. Administrators can send an earlier or a later software version than the current version on the phone.

User-controlled updates apply to configuration changes and software updates you make on the server and Web Configuration Utility. If a user postpones a software update, configuration changes and software version updates from both the server and Web Utility are postponed. When the user chooses to update, configuration and software version changes from both the server and Web Utility are sent to the phone.

This feature does not work if you have enabled ZTP or Skype for Business Device Update, and is not available with Skype for Business.

Set Software Update Polling Policies

You can set a polling policy and polling time period at which the phone polls the server for software updates and displays a notification on the phone to update software. For example, if you set the polling policy to poll

every four hours, the phone polls the server for new software every four hours and displays a notification letting the user know that a software update is available. Users can choose to update the software or they postpone it to a maximum of three times for up to six hours. The phone automatically updates the software after three postponements or after six hours, whichever comes first.

The polling policy is disabled after the phone displays the software update notification.

After the software postponement ends, the phone displays the software update notification again.

User-Controlled Software Updates and Polling Parameters

Parameter template	Permitted Values
`prov.usercontrol.enabled` site.cfg	1 (default) - The phone displays the software update notification and options and the user can control the software download.
	0 - The phone does not display the software update notification and options and the phone reboots automatically to update the software.
`prov.usercontrol.postponeTime` site.cfg	Configure a time interval for software update notications using the format HH:MM. If you configures an invalid value the default value is used.
	2 hours (default), 15 min, 1 hour, 2hours, 4 hours, 6 hours.

Update to a Previous UC Software Release

If you are using UC Software 4.0.0 or later and want to update to a previous software release, follow the instructions in *Upgrading Polycom Phones to and Downgrading Phones from Polycom UC Software 4.0.0*: T*echnical Bulletin 64731 at* Polycom Engineering Advisories and Technical Notifications.

Update Phones from UC Software 4.0.x

If your Polycom phones are running UC Software 4.0.x or later, you can update to a later UC Software version using the instructions in this section. If your phones are running a software release earlier than UC Software 4.0.x, you can update to UC Software 4.0.x following the instructions in *Technical Bulletin 64731: Upgrading Polycom Phones to and Downgrading Phones From Polycom UC Software 4.0.0* at Polycom Engineering Advisories and Technical Notifications.

> **Caution: Mandatory changes to configuration files**
> To ensure predictable phone behavior, the configuration files listed in CONFIG_FILES attribute of the master configuration file must be updated when the software is updated. For information on this attribute refer to Master Configuration File Fields.

To update phones to a version later than Polycom UC Software 4.0.1:

1 Back up your existing application and configuration files.

2 Create your new configuration using UC Software 4.1.0.

   Configuration file changes and enhancements are explained in the Release Notes that accompany the software.

3 Save the new configuration files and images (such as sip.ld) on your provisioning server.

4 Reboot the phones using an automatic method such as polling or check-sync.

  ➢ Reboot your phone manually as a backup option only if another reboot method fails. For details on manually rebooting a phone, refer to Use Multiple Key Combinations.

  ➢ You can boot the phones remotely through the SIP signaling protocol. Refer to <voIpProt.SIP.specialEvent.*/>.

You can configure the phones to periodically poll the provisioning server for changed configuration files or application executables. If a change is detected, the phone may reboot to download the change.

Trusted Certificate Authority List

Polycom maintains and publishes a list of trusted certificate authorities (CAs) supported by each major Polycom UC Software release. To find the list of supported CAs for your UC Software version, see *Certificate Updates for Polycom UC Software – Technical Update* for your UC Software version at Voice Support. Polycom publishes the following details for each trusted CA:

● Certificate Common Name (CN)

● RSA public key size

● Signature algorithm

● Start and end date of certificate validity

Polycom makes every effort to maintain a built-in list of the most commonly used Certificate Authority (CA) certificates. Due to memory constraints, we cannot ensure a complete set of certificates.

If you are using a certificate from a commercial CA not currently support, you can submit a feature request for Polycom to add your CA to the trusted list. You can also load your particular CA certificate into the phone using the custom certificate method shown in *Using Custom Certificates on Polycom Phones - Technical Bulletin 17877* at Polycom Engineering Advisories and Technical Notifications.

OpenSSL Versions List

To view release notes for all Open SSL versions, see OpenSSL Release Notes.

OpenSSL Versions

UC Software Version	OpenSSL Version
UC Software 5.4.1	OpenSSL 1.0.1p 9 July 2015
UC Software 5.4.0	OpenSSL 1.0.1p 9 July 2015
UC Software 5.3.0	OpenSSL 1.0.1j 15 Oct 2014
UC Software 5.2.2	OpenSSL 1.0.1j 15 Oct 2014
UC Software 5.2.0	OpenSSL 1.0.1h 5 Jun 2014
UC Software 5.1.3	OpenSSL 1.0.1h 5 Jun 2014
UC Software 5.1.2	OpenSSL 1.0.1h 5 Jun 2014
UC Software 5.0.2	OpenSSL 1.0.1c 10 May 2012

OpenSSL Versions

UC Software Version	OpenSSL Version
UC Software 5.0.1	OpenSSL 1.0.1c 10 May 2012
UC Software 5.0.0	OpenSSL 1.0.1c 10 May 2012

Encrypt Configuration Files

Polycom phones can download encrypted files from the provisioning server and encrypt files before uploading them to the provisioning server. You can encrypt all configuration files except the master configuration file, contact directory files, and configuration override files from the Web Configuration Utility and local device interface.

To encrypt files, you must provide the phone an encryption key. You can generate your own 32 hex-digit, 128 bit key or use the Polycom 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. Note that the SDK generates a random key and applies Advanced Encryption Standard (AES) 128 in Cipher Block Chaining (CBC) mode, for example:

```
Crypt=1;KeyDesc=companyNameKey1;Key=06a9214036b8a15b512e03d53412006;
```

> **Web Info: Using the SDK to encrypt files**
> To request the SDK and quickly install the generated key, see *When Encrypting Polycom UC Software Configuration Files*: *Quick Tip 67442 at* Polycom Engineering Advisories and Technical Notifications.

You can use the following parameters to set the key on the phone:

- `device.set`
- `device.sec.configEncryption.key`
- `device.sec.configEncryption.key.set`

If the phone doesn't have a key, you must download the key to the phone in plain text, which is a potential security concern if you are not using HTTPS. If the phone already has a key, you can download a new key. Polycom recommends naming each key uniquely to identify which key was used to encrypt a file.

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**.

> **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 continues to do this until the provisioning server provides an encrypted file that can be read, an unencrypted file, or until the file is removed from the list in the master configuration file.

Change the Encryption Key on the Phone and Server

To maintain secure files, you can change the encryption key on the phones and the server.

To change an encryption key on the phone:

1 Place all encrypted configuration files that you want to use the new key on the provisioning server.

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 Provision the phone again so that it downloads the new key. The phone automatically reboots 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 replaces them when it successfully boots.

Check an Encrypted File

You can check whether or not an encrypted file and an unencrypted file are the same.

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, which shows the "digest" field.

2 View the encrypted file with text editor, and check the Digest=…." field. If the two fields are the same, then the encrypted and unencrypted file are the same.

Reboot the Phone

You can reboot the phone from the phone menu when you want to send configuration changes requiring a reboot or restart to the phone. Parameters that require a reboot or restart are marked in the parameter tables in this guide. For configuration changes that do not require a reboot or restart, you can update configuration.

To reboot the phone from the phone menu:

» On the phone, go to **Settings > Advanced > Reboot Phone**.

Restart the Phone

You can restart the phone from the phone menu when you want to send configuration changes requiring a reboot or restart to the phone. Parameters that require a reboot or restart are marked in the parameter tables in this guide. For configuration changes that do not require a reboot or restart, you can update configuration.

To restart the phone from the phone menu:

» On the phone, go to **Settings > Basic > Update Configuration**.

If new Updater or Polycom UC Software is available on the provisioning server, the phone downloads the software. If new software is available on the provisioning server, the phone downloads the software and restarts.

Update Configuration from the Phone Menu

You can update the phone configuration from the phone menu when you want to send configuration changes to the phone. Some configuration changes require a reboot or restart and parameters that require a reboot or restart are marked in the parameter tables in this guide. If there are configuration file changes or new software available on the provisioning server, your phone restarts or reboots if required.

To update configuration from the phone menu:

» On the phone, go to **Settings > Basic** > **Update Configuration**.

Reset the Phone and Configuration

You can reset part or all of the phone and phone configuration.

To access phone reset options:

1 On the phone, go to **Settings > Advanced > Administration Settings > Reset to Defaults**.

The following table describes the phone reset options.

Phone Reset Options

Setting	Description
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 re-downloads 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.

Use Multiple Key Combinations

You can reboot the phone, reset the phone to factory default values, upload log files from the phone to your provisioning server, set the Base Profile, and view phone details with a multiple key combination (MKC) on your Polycom phones.

Web Info: Resetting and rebooting your phone

For other methods for resetting and rebooting your Polycom phones, refer to *Updating, Troubleshooting, and Resetting SoundPoint IP, SoundStation IP, and VVX 1500 Phones*: *Quick Tip 18298 at* Polycom Engineering Advisories and Technical Notifications.

Reboot the Phone with an MKC

You can reboot the phones with a multiple key combination that varies by phone model. Depending on your phone model, press and hold the following keys simultaneously until you hear a confirmation tone (for about three seconds). Rebooting the phone downloads new software and new configuration files if available on the provisioning server.

Phone Reboot Multiple Key Combinations

Phone Model	MKC
VVX 101, 201	0, 1, and 3
VVX 300, 310	0, 1, and 3
VVX 301, 311	0, 1, and 3
VVX 400, 410	0, 1, and 3
VVX 401, 411	0, 1, and 3
VVX 500, 501	0, 1, and 3
VVX 600, 601	0, 1, and 3
VVX 1500	Delete, Volume-, Volume+, and Select

Reset the Phone to Defaults with an MKC

You can reset a phone to factory default settings with a multiple key combination that varies by phone model. This is useful when you use more than one method to configure phones and phone features. Resetting the phone to defaults clears the flash parameters, removes log files, user data, and cached data, and resets the administrator password to 456.

Depending on your phone model, press and hold the following keys simultaneously during the updater/BootROM countdown process until the administrator password prompt displays.

Resetting to factory defaults also resets the administrator password (factory default password is 456). Polycom recommends that you change the administrative password from the default value.

Settings: Resetting a VVX 1500 D to default values disables the H.323 protocol

After you reset to factory defaults on a Polycom VVX 1500 D phone, you must re-enable the H.323 protocol through a configuration file change or by using the Web Configuration Utility. Refer to the section Configure H.323 Protocol.

Factory Default Multiple Key Combinations

Phone Model	MKC
VVX 101, 201	*, #
VVX 300, 310	*, #
VVX 301, 311	*, #
VVX 400, 410	*, #
VVX 401, 411	*, #
VVX 500, 501	*, #
VVX 600, 601	*, #
VVX 1500	4, 6, 8, and * dial pad keys

Upload Log Files with an MKC

You can use a multiple key press to upload log files to your provisioning server with a multiple key combination that varies by phone model. Uploading log files copies the log files from the phone to the provisioning server. and creates new files named *<MACaddress>-now-xxx.log*. Depending on your phone model, press and hold one the following keys simultaneously for about three seconds until you hear a confirmation tone.

Factory Default Multiple Key Combinations

Phone Model	MKC
VVX 101, 201	1, 5, and 9
VVX 300, 310	1, 5, and 9
VVX 301, 311	1, 5, and 9
VVX 400, 410	1, 5, and 9
VVX 401, 411	1, 5, and 9
VVX 500, 501	1, 5, and 9
VVX 600, 601	1, 5, and 9
VVX 1500	Up, Down, Left, and Right arrow keys

Set the Base Profile with an MKC

Setting the base profile allows for quick setup of Polycom phones with Microsoft Lync Server and Skype for Business Server. Depending on your phone model, press and hold the following keys simultaneously for about three seconds until you hear a confirmation tone.

Factory Default Multiple Key Combinations

Phone Model	MKC
VVX 101, 201	1, 4, and 9
VVX 300, 310	1, 4, and 9
VVX 301, 311	1, 4, and 9
VVX 400, 410	1, 4, and 9
VVX 401, 411	1, 4, and 9
VVX 500, 501	1, 4, and 9
VVX 600, 601	1, 4, and 9
VVX 1500	1, 4, and 9

View Phone Details with an MKC

You can use a multiple key combination to view frequently-used administrator phone details including:

- IP Address
- Boot Server Type
- MAC Address
- VLAN
- Boot Server Address
- UC Software version

To view VVX phone details on the phone:

1 Press and hold keys **1**,**4**, and **7**.

Define the Phone Key Layout

You can redefine certain hard key functions using parameters in the configuration files. The following figures and tables show the default key layouts for the following phones:

- VVX 101 and 201 Default Phone Key Functions
- VVX 300 Series Default Phone Key Functions
- VVX 400 Series Default Phone Key Functions
- VVX 500 Series Default Phone Key Functions
- VVX 600 Series Default Phone Key Functions
- VVX 1500 Default Phone Key Functions

VVX 101 and 201

The following figure and table show the available phone key functions.
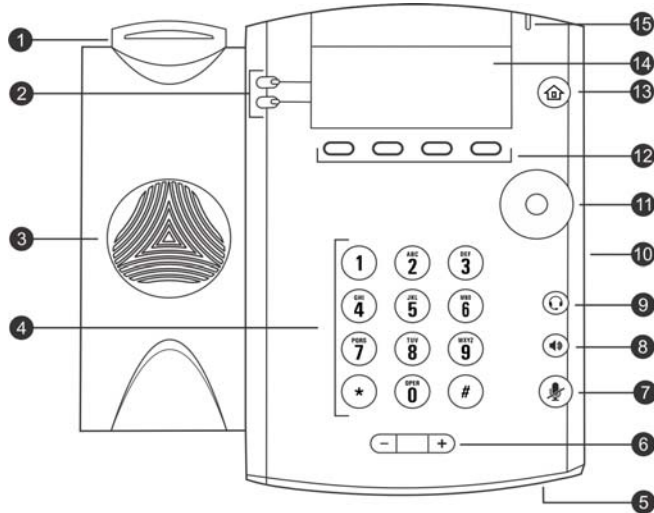
VVX 101 and 201 Default Phone Key Functions

KEY ID	Function	KEY ID	Function
1	Hookswitch	9	Headset key
2	Line keys	10	Security slot (on side)
3	Speaker	11	Navigation keys / Select key
4	Dial pad keys	12	Soft keys
5	Microphone	13	Home key
6	Volume keys	14	Screen
7	Mute key	15	Message Waiting Indicator
8	Speakerphone key		

VVX 300 Series

The following figure and table show the available phone key functions. IDs that have no function are described as n/a.

Key ID

VVX 300 Series Default Phone Key Functions

KEY ID	Function	KEY ID	Function	KEY ID	Function	KEY ID	Function
1	n/a	15	Dialpad7	29	SoftKey1	43	n/a
2	Dialpad2	16	Dialpad8	30	n/a	44	n/a
3	ArrowLeft	17	Dialpad9	31	SoftKey4	45	n/a
4	ArrowRight	18	Select	32	Line2	46	n/a
5	Dialpad3	19	Hold	33	Line3	47	n/a
6	VolDown	20	Transfer	34	Line4	48	n/a
7	VolUp	21	Messages	35	n/a	49	n/a
8	Dialpad4	22	DialpadStar	36	n/a	50	n/a
9	Dialpad5	23	Dialpad0	37	n/a	51	Line1
10	Headset	24	DialpadPound	38	n/a	52	Line5
11	ArrowDown	25	Dialpad1	39	n/a	53	Line6
12	ArrowUp	26	Home	40	Dialpad6		

VVX 300 Series Default Phone Key Functions

KEY ID	Function	KEY ID	Function	KEY ID	Function	KEY ID	Function
13	Handsfree	27	SoftKey3	41	n/a		
14	MicMute	28	SoftKey2	42	n/a		

VVX 400 Series

The following figure and table show the available phone key functions. IDs that have no function are described as n/a.

⬤ Key ID

VVX 400 Series Default Phone Key Functions

KEY ID	Function	KEY ID	Function	KEY ID	Function	KEY ID	Function
1	n/a	15	Dialpad7	29	SoftKey1	43	n/a
2	Dialpad2	16	Dialpad8	30	n/a	44	n/a
3	ArrowLeft	17	Dialpad9	31	SoftKey4	45	n/a
4	ArrowRight	18	Select	32	Line2	46	n/a
5	Dialpad3	19	Hold	33	Line3	47	n/a
6	VolDown	20	Transfer	34	Line4	48	n/a
7	VolUp	21	Messages	35	Line8	49	n/a
8	Dialpad4	22	DialpadStar	36	Line9	50	n/a
9	Dialpad5	23	Dialpad0	37	Line10	51	Line1
10	Headset	24	DialpadPound	38	n/a	52	Line5
11	ArrowDown	25	Dialpad1	39	n/a	53	Line6
12	ArrowUp	26	Home	40	Dialpad6	54	Line7

VVX 400 Series Default Phone Key Functions

KEY ID	Function	KEY ID	Function	KEY ID	Function	KEY ID	Function
13	Handsfree	27	SoftKey3	41	n/a	55	Line11
14	MicMute	28	SoftKey2	42	n/a	56	Line12

VVX 500 Series

The following figure and table show the available phone key functions. IDs that have no function are described as n/a.

VVX 500 Series Default Phone Key Functions

KEY ID	Function	KEY ID	Function	KEY ID	Function	KEY ID	Function
1	Dialpad1	12	Headset	23	Dialpad0	34	n/a
2	Dialpad2	13	n/a	24	DialpadPound	35	n/a
3	VolDown	14	n/a	25	n/a	36	n/a
4	VolUp	15	Dialpad7	26	Home	37	n/a
5	Dialpad3	16	Dialpad8	27	n/a	38	n/a
6	n/a	17	Dialpad9	28	n/a	39	n/a
7	n/a	18	MicMute	29	n/a	40	Dialpad6
8	Dialpad4	19	n/a	30	n/a	41	n/a
9	Dialpad5	20	n/a	31	n/a	42	n/a
10	n/a	21	n/a	32	n/a		
11	Handsfree	22	DialpadStar	33	n/a		

VVX 600 Series

The following figure and table show the available phone key functions. IDs that have no function are described as n/a.
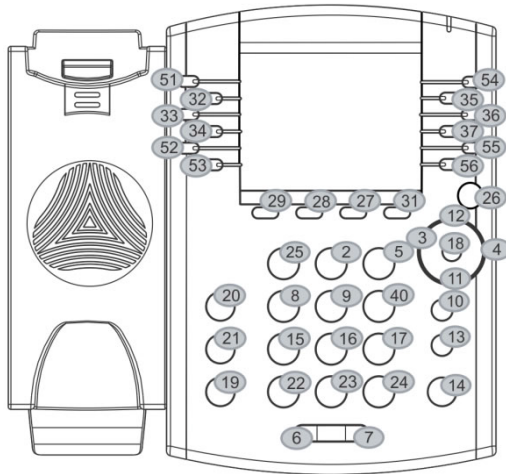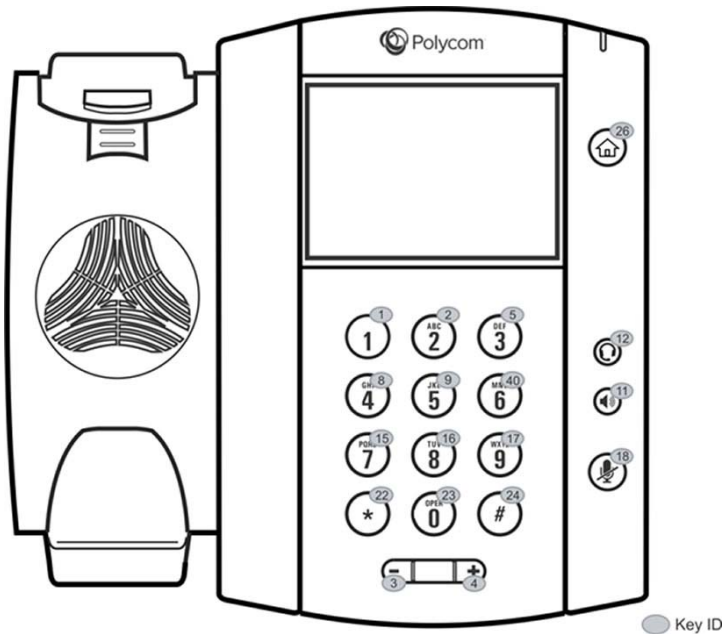

VVX 600 Series Default Phone Key Functions

KEY ID	Function	KEY ID	Function	KEY ID	Function	KEY ID	Function
1	Dialpad1	12	Headset	23	Dialpad0	34	n/a
2	Dialpad2	13	n/a	24	DialpadPound	35	n/a
3	VolDown	14	n/a	25	n/a	36	n/a
4	VolUp	15	Dialpad7	26	Home	37	n/a
5	Dialpad3	16	Dialpad8	27	n/a	38	n/a
6	n/a	17	Dialpad9	28	n/a	39	n/a
7	n/a	18	MicMute	29	n/a	40	Dialpad6
8	Dialpad4	19	n/a	30	n/a	41	n/a
9	Dialpad5	20	n/a	31	n/a	42	n/a
10	n/a	21	n/a	32	n/a		
11	Handsfree	22	DialpadStar	33	n/a		

VVX 1500

The following figure and table show the available phone key functions. IDs that have no function are described as n/a.
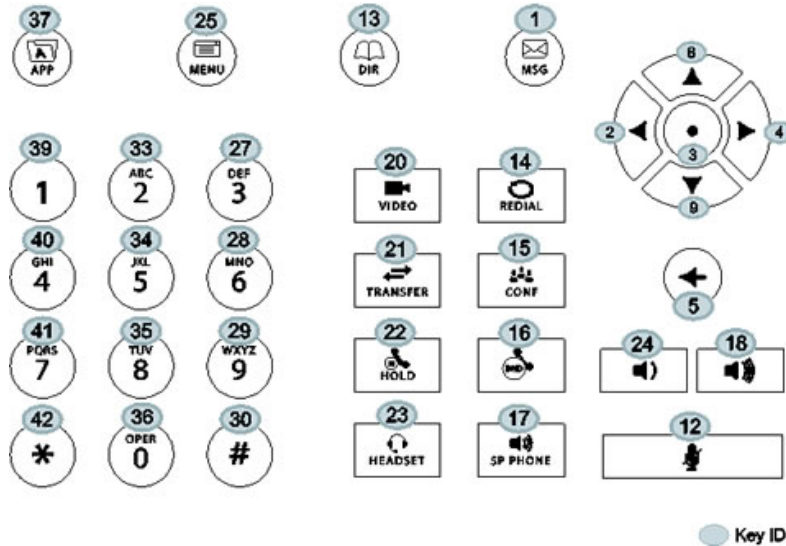
VVX 1500 Default Phone Key Functions

KEY ID	Function	KEY ID	Function	KEY ID	Function	KEY ID	Function
1	Messages	12	MicMute	23	Headset	34	Dialpad5
2	ArrowLeft	13	Directories	24	VolDown	35	Dialpad8
3	Select	14	Redial	25	Menu	36	Dialpad0
4	ArrowRight	15	Conference	26	n/a	37	Applications
5	Delete	16	DoNotDisturb	27	Dialpad3	38	n/a
6	n/a	17	Handsfree	28	Dialpad6	39	Dialpad1
7	n/a	18	VolUp	29	Dialpad9	40	Dialpad4
8	ArrowUp	19	n/a	30	DialpadPound	41	Dialpad7
9	ArrowDown	20	Video	31	n/a	42	DialpadStar
10	n/a	21	Transfer	32	n/a		
11	n/a	22	Hold	33	Dialpad2		

Map 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.

Note the following guidelines:

- 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.
- Some functions depend on the feature being enabled. For example, the BuddyStatus and MyStatus soft keys require the presence feature to be enabled.
- Hard key remappings do not require the enhanced feature key feature to be enabled. This includes the speed dial function on older platforms. On newer platforms, use line key functions.

The table below shows only line1 to line 6 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	
ArrowDown	Move arrow down	
ArrowLeft	Move arrow left	
ArrowRight	Move arrow right	
ArrowUp	Move arrow up	
BargeIn	Barge In to show appearances, Barge In	Call screen only
BuddyStatus	Buddy Status	
Callers	Callers	
CallList	Call Lists	
CallPark	ParkEntry	Call screen only
CallPickup	Pickup a call	Call screen only
Conference	Begin a conference call	Call screen only
Delete	Delete	
Dialpad0	Dialpad 0	
Dialpad1	Dialpad 1	
Dialpad2	Dialpad 2	

Key Labels and Internal Functions

Function	Description	Notes
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
DirectedPickup	Directed pickup	Call screen only
Directories	Directories	
Divert	Forward	
DoNotDisturb	Do Not Disturb menu	
EnterRecord	Enter a call record	Call screen only
Exit	Exit existing menu	Menu only
GroupPickup	Group pickup	
Handsfree	Use handsfree	
Headset	Use headset	Desktop phones only
Hold	Toggle hold	
Join	Join	Call screen only
LCR	Last call return	
Line1	Line Key 1	
Line2	Line Key 2	
Line3	Line Key 3	
Line4	Line Key 4	
Line5	Line Key 5	
Line6	Line Key 6	
ListenMode	Turn on speaker to listen only	

Key Labels and Internal Functions

Function	Description	Notes
LockPhone	Lock the phone	
Menu	Menu	
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	
ParkedPickup	Parked pickup	Call screen only
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	
Softkey5	Softkey 5	
SpeedDial	SpeedDial	
Split	Split	Call screen only
Talk	Push-to-Talk	
Transfer	Transfer	Call screen only
Video	Video	Polycom VVX 500/501, 600/601, and 1500

Key Labels and Internal Functions

Function	Description	Notes
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 virtual local area network (VLAN) statically in the phone's network configuration menu or use Cisco Discovery Protocol (CDP) or Link-Layer Discovery Protocol (LLDP) 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 to understand the phone boot-up sequence when assigning a VLAN ID via DHCP.

VLAN using DHCP phone boot-up sequence
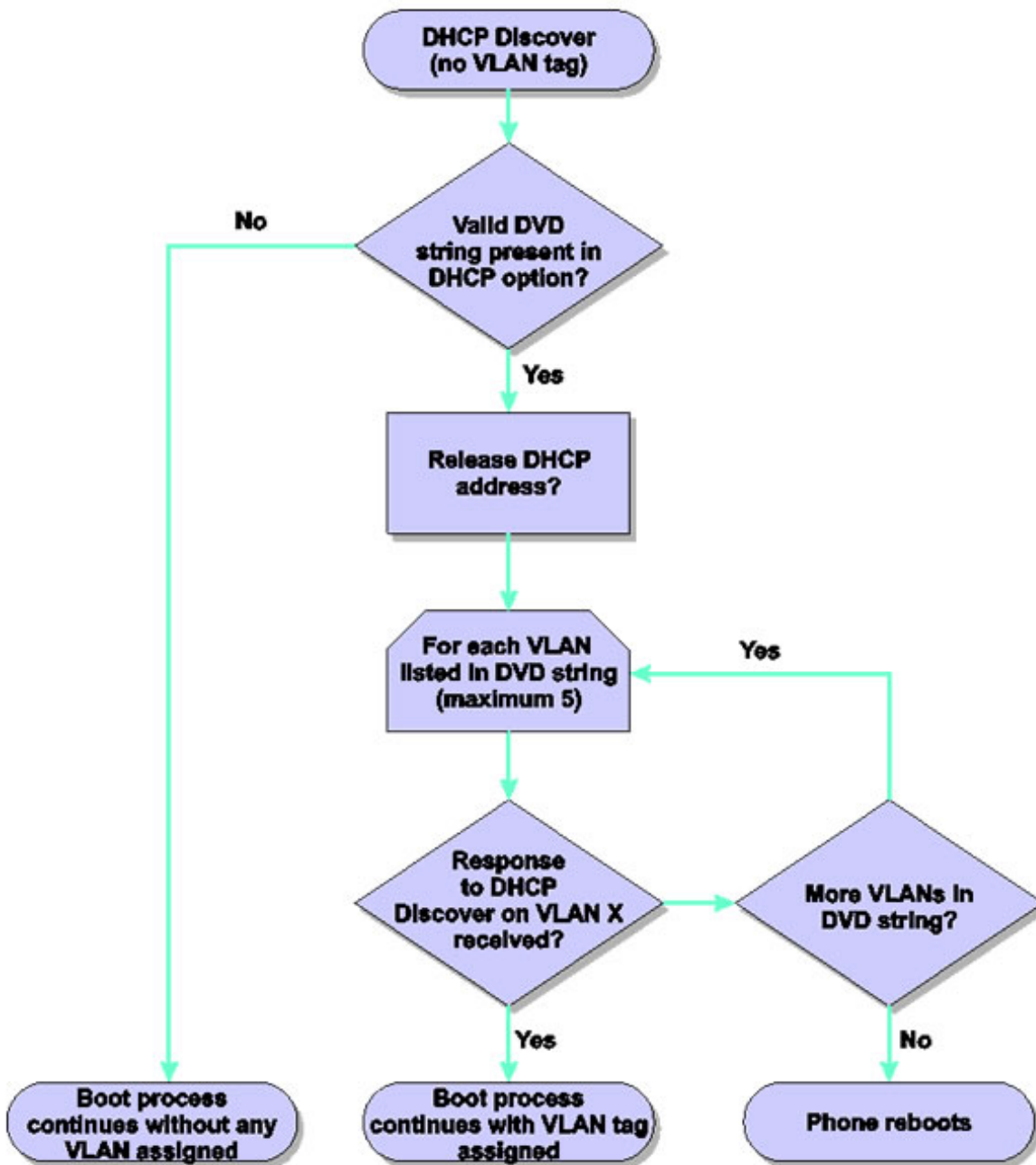
To assign a VLAN ID to a phone using DHCP:

» In the DHCP menu of the Main setup menu, set **VLAN Discovery** to **Fixed** or **Custom**.

When set to Fixed, the phone examines 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 are 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 ";" are ignored

● There must be no white space before the semi colon ";"

● VLAN IDs may be decimal, hex, or octal

The following DVD strings 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 are ignored.

Parse Vendor ID Information

After the phone boots up, it sends a DHCP discover packet to the DHCP server. The DHCP discover packet is located 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.

Polycom follows RFC 3925 which specifies use of a unique IANA private enterprise number. The private enterprise number assigned to Polycom is 13885 (0x0000363D) represented as an array of binary data.

To parse vendor ID information:

1 Check for the Polycom signature at the start of the option: `4 octet: 00 00 36 3d`

2 Obtain 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).

The following example is a sample decode of a packet (DHCP Option 60) from a VVX 500/501:

```
3c 7a
```

➢ Option 60, length of Option data (part of the DHCP specification)

```
00 00 36 3d
```

➢ Polycom signature (always 4 octets)

```
75
```

➢ Length of Polycom data

```
01 07 50 6f 6c 79 63 6f 6d
```

➢ sub-option 1 (company), length, "Polycom"

```
02 0b 56 56 58 2d 56 56 58 5f 34 31 30
```

➢ sub-option 2 (part), length, "VVX-VVX_500/501"

```
03 10 33 31 31 31 2d 34 36 31 36 32 2d 30 30 31 2c 37
```

➢ sub-option 3 (part number), length, "3111-44500-001,7"

```
04 1e 53 49 50 2f 35 2e 32 2e 30 2e 58 58 58 58 2f 30 36 2d 41 75 67 2d 31 34
20 32 30 3a 35 35
```

➢ sub-option 4 (Application version), length, "SIP/5.2.0.XXXX/06-Aug-14 20:55"

```
05 1d 55 50 2f 35 2e 34 2e 30 2e 58 58 58 58 2f 30 36 2d 41 75 67 2d 31 34 20
32 31 3a 30 34
```

➢ sub-option 5 (Updater version), length, "UP/5.4.0.XXXX/06-Aug-14 21:04"

```
06 0c 64 73 6c 66 6f 72 75 6d 2e 6f 72 67
```

➢ sub-option 6 "dslforum.org"

Disable the PC Ethernet Port

You can disable the Ethernet port and the PC Ethernet port on all devices from the phone interface.

To disable Ethernet:

1 Navigate to the phone's Ethernet Menu (**Menu > Settings > Advanced** (default password 456) > **Administration Settings > Network Configuration > Network Interfaces > Ethernet Menu**).
2 Scroll down to **PC Port Mode** and press the **Edit** soft key.
3 Select **Disabled** and press the **OK** soft key.
4 Press the **Exit** soft key and select **Save Config**.

The phone reboots. When the reboot is complete, the PC Ethernet port is disabled.

Capture Your Device's Current Screen

You can capture your phone or expansion module's current screen. Note that you must enable the phone's web server using the parameter `httpd.enabled` before you can take a screen capture. To capture the current screen of expansion modules, you must provide power and connect the expansion module to a phone.

To capture a device's current screen:

1 In the `sip-interop.cfg` template, locate the parameter `up.screenCapture.enabled`.

You can add the sip-interop.cfg template to the CONFIG-FILES field of the master configuration file, or copy the parameter to an existing configuration file.

2 Set the value to `1` and save the configuration file.

3 On the device, go to **Settings > Basic > Preferences > Screen Capture**.

Note you must repeat step 3 each time the device restarts or reboots.

4 Locate and record the phone's IP address at **Status > Platform > Phone > IP Address**.

5 Set the phone to the screen you want to capture.

6 In a web browser address field, enter `https://<phoneIPaddress>/captureScreen` where `<phoneIPaddress>` is the IP address you obtained in step 5.

The web browser displays an image showing the phone's current screen. You can save 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 LLDP protocol was formally ratified as IEEE standard 802.1AB in May 2005. Refer to section 10.2.4.4 of the LLDP-MED standard. Note also that the standard 802.3at-2009 extended LLDP definitions.

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 methods
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. Static (the VLAN ID is entered through the phone's user interface); 4. DVD (VLAN Via DHCP).

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 is recognized as LLDP only if it contains mandatory TLVs. Polycom phones running the UC Software 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

According to section 10.2.4.4 of the LLDP-MED standard, LLDP-MED devices must 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. As a result, Polycom phones do 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.

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	-	-

Supported TLVs

No	Name	Type (7 bits) [0-6]	Length (9 bits) [7-15]	Type Length	Org. Unique Code (3 bytes)	Sub Type
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	-	-	-
Refer to System and Model Names.						
6	System description	6	min len > 0, max len <= 255	-	-	-
Manufacturer's name - "Polycom"; Hardware version; Application version; 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: After 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						
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

Supported TLVs

No	Name	Type (7 bits) [0-6]	Length (9 bits) [7-15]	Type Length	Org. Unique Code (3 bytes)	Sub Type

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.

No	Name	Type	Length	Type Length	Org. Unique Code	Sub Type
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: VVX 500/501, 600/601, and 1500 business media phones.

No	Name	Type	Length	Type Length	Org. Unique Code	Sub Type
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 - Refer to Phone Power Values.

| 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)

| 20 | LLDP-MED inventory manufacturer name | 127 | 11 | 0xfe0b | 0x0012bb | 9 |

Polycom

Supported TLVs

No	Name	Type (7 bits) [0-6]	Length (9 bits) [7-15]	Type Length	Org. Unique Code (3 bytes)	Sub Type
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.

System and Model Names

The following table outlines Polycom phone models, and their system and model names.

Phone System and Model Names

Model	System Name	Model Name
VVX 101	Polycom VVX 101	VVX-VVX_101
VVX 201	Polycom VVX 201	VVX-VVX_201
VVX 300	Polycom VVX 300	VVX-VVX_300
VVX 301	Polycom VVX 301	VVX-VVX_301
VVX 310	Polycom VVX 310	VVX-VVX_310
VVX 311	Polycom VVX 311	VVX-VVX_311
VVX 400	Polycom VVX 400	VVX-VVX_400
VVX 401	Polycom VVX 401	VVX-VVX_401
VVX 410	Polycom VVX 410	VVX-VVX_410
VVX 411	Polycom VVX 411	VVX-VVX_411
VVX 500	Polycom VVX 500	VVX-VVX_500
VVX 501	Polycom VVX 501	VVX-VVX_501
VVX 600	Polycom VVX 600	VVX-VVX_600
VVX 601	Polycom VVX 601	VVX-VVX_601

Phone System and Model Names

Model	System Name	Model Name
VVX 1500	Polycom VVX 1500	VVX-VVX_1500
SoundStructure	SoundStructure VoIP Interface	SoundStructure VoIP Interface

PMD Advertise and Operational MAU

The following table lists values for the PMD advertise and operational MAU.

By default, all Polycom phones have the PMD advertise capability set for 10HD, 10FD, 100HD, and 100FD bits. The VVX 310/311, 410/411, 500/501, 600/601, and 1500 business media phones have Gigabit Ethernet and support the PMD advertise capability set for 1000FD bit.

PMD Advertise and Operational 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

Phone Power Values

The following table outlines the power usage for each phone, as well as the power value sent in LLDP-MED.

Power Consumption – Network Standby

In accordance with section 7 of the EU regulation 801/2013, Polycom provides the power consumption figures for its VoIP telephones when in their network standby state. To view this information, see Polycom Environment Compliance.

Phone Power Values

Model	Power Usage (Watts)	Power Value Sent in LLDP-MED Extended Power Via MDI TLV (Watts)
VVX 101	5.0	50
VVX 201	5.0	50

Phone Power Values

Model	Power Usage (Watts)	Power Value Sent in LLDP-MED Extended Power Via MDI TLV (Watts)
VVX 300	5.0	50
VVX 301	5.0	50
VVX 310	5.0	50
VVX 311	5.0	50
VVX 400	5.0	50
VVX 401	5.0	50
VVX 410	5.0	50
VVX 411	5.0	50
VVX 500	8.0	80
VVX 501	8.0	80
VVX 600	8.0	80
VVX 601	8.0	80
VVX 1500	11.8	118
SoundStructure	4.78	na