ADMINISTRATOR GUIDE

UC Software 5.4.4 | December 2016 | 3725-20727-005A

Polycom® RealPresence Trio™ Solution

6001 America Center Drive
San Jose, CA 95002
USA

Trademarks Polycom®, the Polycom logo and the names and marks associated with Polycom products are trademarks and/or service marks of Polycom, Inc. and are registered and/or common law marks in the United States and various other countries.

All other trademarks are property of their respective owners. No portion hereof may be reproduced or transmitted in any form or by any means, for any purpose other than the recipient's personal use, without the express written permission of Polycom.

Disclaimer While Polycom uses reasonable efforts to include accurate and up-to-date information in this document, Polycom makes no warranties or representations as to its accuracy. Polycom assumes no liability or responsibility for any typographical or other errors or omissions in the content of this document.

Limitation of Liability Polycom and/or its respective suppliers make no representations about the suitability of the information contained in this document for any purpose. Information is provided "as is" without warranty of any kind and is subject to change without notice. The entire risk arising out of its use remains with the recipient. In no event shall Polycom and/or its respective suppliers be liable for any direct, consequential, incidental, special, punitive or other damages whatsoever (including without limitation, damages for loss of business profits, business interruption, or loss of business information), even if Polycom has been advised of the possibility of such damages.

End User License Agreement By installing, copying, or otherwise using this product, you acknowledge that you have read, understand and agree to be bound by the terms and conditions of the End User License Agreement for this product. The EULA for this product is available on the Polycom Support page for the product.

Patent Information The accompanying product may be protected by one or more U.S. and foreign patents and/or pending patent applications held by Polycom, Inc.

Open Source Software Used in this Product This product may contain open source software. You may receive the open source software from Polycom up to three (3) years after the distribution date of the applicable product or software at a charge not greater than the cost to Polycom of shipping or distributing the software to you. To receive software information, as well as the open source software code used in this product, contact Polycom by email at OpenSourceVideo@polycom.com.

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

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

Contents

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 RealPresence Trio 8800 and RealPresence Trio Visual+ systems

Web Info: Latest Polycom UC Software for RealPresence Trio solution

To find out what's new for this release of UC Software, including enhanced features, and known and resolved issues, see the release notes at RealPresence Trio 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:

Provisioning and Configuring 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
- Setting 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 the following functions and features on the phones:

- VoIP signaling for a wide range of voice and video telephony functions using SIP signaling for call setup and control.
- SIP signaling 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 USB 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	
`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
`region.cfg`	Non-North American geographies	Typical Hosted Service Provider Typical IP-PBX

Configuration File Templates

Name	Description	Deployment Scenarios
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 value types. 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 interchangeable.
- The values `1, true`, and `on` are supported and interchangeable. 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, and 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 `&amp;`
- ” as `&quot;`
- ’ as `&apos;`
- < as `&lt;`
- > as `&gt;`
- 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 Setting 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
- USB bulk provisioning
- Polycom® RealPresence® Resource Manager software
- 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 button displays on the phone interface when the phone is booting. This button allows administrators or users to access the provisioning server and configure the phone for provisioning. For more detail details on how to configure quick setup, see Technical Bulletin 45460: Using Quick Setup with Polycom Phones.

After the initial configuration is complete, you can show or hide the QSetup button using the parameter in the following table.

Configuring Quick Setup

Use the parameters in this section to 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 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. Some settings require a device restart or reboot.

If you want to reset all settings made from the RealPresence Trio 8800 menu to default or reset the device to factory defaults, refer to Restart, Reset to Defaults, Upload Log Files.

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.

USB Provisioning

You can provision RealPresence Trio 8800 or RealPresence Trio Visual+ with configuration files stored on a USB device during normal functioning or in recovery mode. Recovery mode enables you to recover the RealPresence Trio 8800 or RealPresence Trio Visual+ to a normal provisioning state when other methods are not working or not available.

RealPresence Trio 8800 supports only File Allocation Table (FAT) file systems and Polycom recommends using FAT32.

If other USB devices are attached to RealPresence Trio 8800, you must remove them and ensure that RealPresence Trio 8800 correctly recognizes the USB device you want to install from.

If you use a USB device to provision while centralized provisioning server is in use, the USB configuration files override server settings. When you remove the USB device, the device returns to settings you configured on the server. Note, however, that the original server settings are subject to direct.set changes initiated by the USB device. The direct.set changes can alter parameters on the provisioning server and change basic provisioning settings.

When you attach a USB device, you are prompted for an administrator password. RealPresence Trio 8800 downloads and installs the configuration files and you can remove the USB when complete.

Update UC Software Manually with a USB Device

You can update UC Software on the RealPresence Trio system manually using a USB device.

To update the software manually with a USB device:

1. Format a USB flash drive as FAT32. Polycom recommends that you use a USB 2.0 flash drive.

   If you are using a drive that is already formatted, ensure that previous files are deleted from the flash drive.
2. Download the UC Software from Polycom Support.
3. Copy the configuration files you want to use to the root of the USB device. The minimum required configuration files must be copied to the are as follows:
   - Master configuration file: `000000000000.cfg`.
   - RealPresence Trio 8800: `3111-65290-001.sip.ld`.
   - RealPresence Trio Visual+: `3111-66420.001.sip.ld`.
4. Connect the USB to the RealPresence Trio 8800 or RealPresence Trio Visual+ USB port.
5. Follow the prompt for the Administrator password.

   The system detects the flash drive and starts the update within 30 seconds. The mute keys' indicator lights begin to flash, indicating that the update has started.

   The system reboots several times during the update. The update is complete when the indicator lights stop flashing and the Home screen displays.

Place the RealPresence Trio Visual+ into Recovery Mode

If other provisioning methods are not working or unavailable, you can place the RealPresence Trio Visual+ into recovery mode to enable you to update software manually using a USB device. Recovery mode places the RealPresence Trio solution to a normal provisioning state.

To place the RealPresence Trio Visual+ into recovery mode:

1. Ensure that the phone is powered off.
2. Plug in a USB device.
3. Power up the phone.
4. When the LED initially turns from on to off, press and hold the pairing button until the pairing LED turns orange and release the button. The pairing LED blinks. (Blinking rotates between orange/red/green/off).

   Recovery process is complete when the device reboots.

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 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.
- You must write the name of configuration files to the CONFIG_FILES field of the master configuration file in the order you want the settings applied. The files you enter to the CONFIG_FILES field 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 configuration setting priority, refer to Polycom Provisioning Methods.

Setting 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, you need to install provisioning tools on your computer and 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.

Install Provisioning Tools

Before you begin provisioning devices with UC Software, install tools on your computer and gather some information.

To install and set up provisioning tools:

1. If using Power over Ethernet (PoE) with the phone, obtain a PoE switch and network cable.
2. Install an XML editor, such as XML Notepad 2007, on your computer.
3. 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.
4. 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.

Set Up a Provisioning Server

After you install provisioning tools, set up the provisioning server.

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.
3. 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.
4. 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.
5. 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.
6. 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. If you set up multiple provisioning servers, you must be able to reach all of the provisioning servers with the same protocol and the contents on each provisioning server must be identical. The default number of provisioning servers is one and the maximum number is eight. For more information on the protocol used, see Supported Provisioning Protocols.

You can 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 try.

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 about the priority of override files, refer to Polycom Provisioning Methods. For information about 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 RealPresence Trio 8800's file system, the override file is deleted. If you need to clear phone settings and features applied by override files, refer to Restart, Reset to Defaults, Upload Log Files.

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: If SNTP settings are not available through DHCP, you may need to edit the SNTP GMT offset or 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 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 **Select**.

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 **Edit** to edit the value and **OK** 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 **Back** twice.

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

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

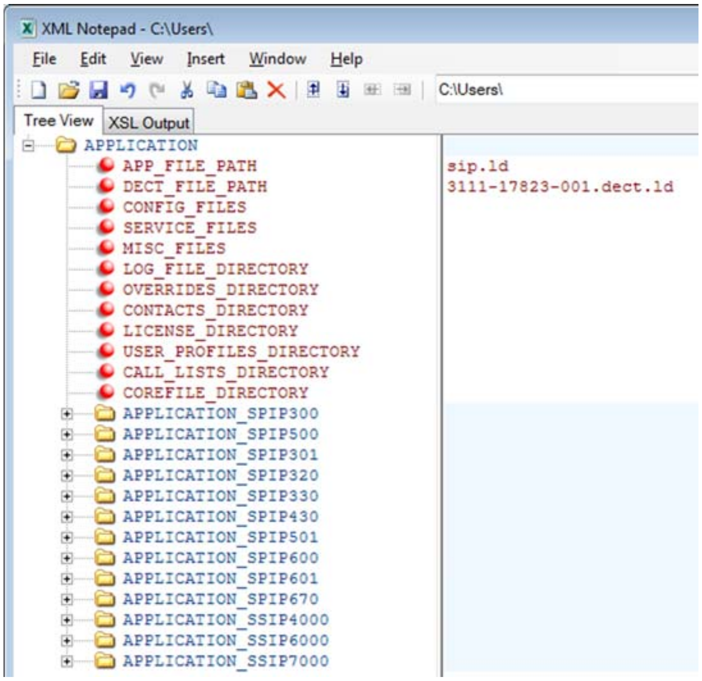
To view phone provisioning information, use the multikey shortcut by simultaneously pressing **1-4-7** to display:

- 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



```
APPLICATION
    APP_FILE_PATH            sip.ld
    DECT_FILE_PATH           3111-17823-001.dect.ld
    CONFIG_FILES
    SERVICE_FILES
    MISC_FILES
    LOG_FILE_DIRECTORY
    OVERRIDES_DIRECTORY
    CONTACTS_DIRECTORY
    LICENSE_DIRECTORY
    USER_PROFILES_DIRECTORY
    CALL_LISTS_DIRECTORY
    COREFILE_DIRECTORY
    APPLICATION_SPIP300
    APPLICATION_SPIP500
    APPLICATION_SPIP301
    APPLICATION_SPIP320
    APPLICATION_SPIP330
    APPLICATION_SPIP430
    APPLICATION_SPIP501
    APPLICATION_SPIP600
    APPLICATION_SPIP601
    APPLICATION_SPIP670
    APPLICATION_SSIP4000
    APPLICATION_SSIP6000
    APPLICATION_SSIP7000
```

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_SPIPXXX 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.

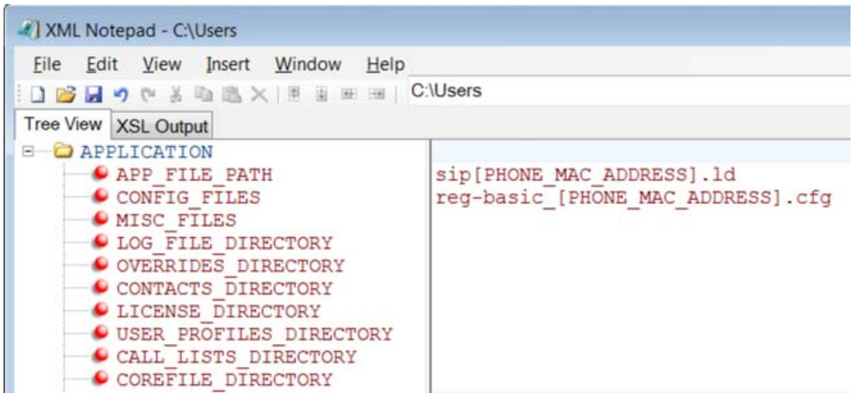
This method is useful if you need to maintain or modify settings common to all of the phones in your deployment. To apply a common configuration to all phones, you need only create one new configuration file and add it to the CONFIG_FILES field of the `000000000000.cfg` master file. If you want to add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



```
APPLICATION
    APP_FILE_PATH              sip[PHONE_MAC_ADDRESS].ld
    CONFIG_FILES               reg-basic_[PHONE_MAC_ADDRESS].cfg
    MISC_FILES
    LOG_FILE_DIRECTORY
    OVERRIDES_DIRECTORY
    CONTACTS_DIRECTORY
    LICENSE_DIRECTORY
    USER_PROFILES_DIRECTORY
    CALL_LISTS_DIRECTORY
    COREFILE_DIRECTORY
```

4. Save the master configuration file.

Configuring 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 the MAC address of a device, refer 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**

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 phones:

1. Create a configuration file with the settings you want to apply. This example uses 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 names, model names, and part numbers for Polycom devices.

Product Name, Model Name, and Part Number

Product Names	Model Names	Part Numbers
RealPresence Trio 8800	Trio8800	3111-65290-001
RealPresence Trio Visual+	TrioVisualPlus	3111-66420-001

Configure Network Settings

The UC Software supports the deployment of RealPresence Trio solution 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.264 Advanced Video Coding (AVC) video endpoint.

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
- 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 Ethernet Settings.
- **VLAN Filtering** To change this parameter, go to the section VLAN Settings.

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 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 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 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 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 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).
- 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.

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 DHCP Settings or Provisioning Server Settings.				
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.				
SNTP GMT offset	2				
Syslog	Refer to the section Syslog Settings				
DNS server IP address	6				

DHCP Network Parameters

Parameter	DHCP Option	DHCP	DHCP INFORM	Configuration File (application only)	Device Settings
DNS INFORM server IP address	6				
DNS domain	15				
VLAN ID	Refer to the section 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.
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	

DHCP Option 43 Configuration Options (continued)

Option	Results
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.

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 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.

Phone Network Settings

You have the option to modify phone network settings. This section lists network settings available from the device interface. 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 **Cancel**, and press **Setup** 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.

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

Main Menu	
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.	

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 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.	
Server Address	**IP address or URL**

Provisioning Server Menu

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.

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**
• **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 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 Provisioning Server Settings. • **Custom + Option 66** The phone uses the custom option first or use Option 66 if the custom option is not present. • Note: If the DHCP server sends nothing, the following scenarios are possible: • If a boot server value is stored in flash memory and the value is not `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. •	
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 Provisioning Server Settings.	

DHCP Menu

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.	

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.

Ethernet Menu

Name	Permitted Values
DHCP	**Enabled, Disabled**
If enabled, DHCP is used to obtain the parameters discussed in the section DHCP Settings.	
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 The phone’s default router.	**IP address**
VLAN See the section 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 802.1X Settings.	
Storm Filtering	**Enabled, Disabled**

Ethernet Menu

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.	
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.	

VLAN Settings

You can modify the settings listed in the following table.

VLAN Menu

Name	Permitted Values
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 Menu

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.	

802.1X Settings

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.	
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.	

Login Credential Settings

You can modify settings shown in the following table.

Login Credentials Menu

Name	Possible Values
Domain The domain name used by a server.	**UTF-8 encoded string**

Login Credentials Menu

User The user name used to authenticate to a server.	**UTF-8 encoded string**
Password The password used to authenticate to a server.	**UTF-8 encoded string**

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 The Online Certificate Status Protocol checks the revocation status of X.509 digital certificates downloaded during negotiation of a TLS connection.	**Enabled, Disabled**
FIPS The Federal Information Processing Standard enables the validation and usage of FIPS-140 certified encryption algorithms.	**Enabled, Disabled**
Install Custom CA Cert A CA certificate that is installed on the phone to be used for TLS authentication.	**URL**
Install Custom Device Cert A device certificate installed on the phone to be used for Mutual TLS authentication.	**URL**
Clear Certificate A flag to determine whether or not the device certificate can be removed from the phone.	**Yes, No**
TLS Profile x There are currently two TLS Platform profiles. See the section TLS Profile Settings.	
Web Services See the section Applications Settings.	

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**

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 The TLS Profile to enable or disable common name validation.	**Enable or Disable**
Syslog The TLS Profile to use for Syslog authentication.	**1 or 2**

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 The syslog server IP address. The default value is Null.	**IP address or domain name string**
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.	

Audio Features

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

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. This feature is enabled by default.

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. This feature is enabled by default.

Synthesized Comfort Noise Fill

This feature is an integral part of handsfree echo reduction and 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. Synthesized comfort noise fill uses noise synthesis techniques to smooth out the noise level in the direction toward the remote user, providing a more natural call experience. This feature is enabled by default.

Synthesized comfort noise fill is unrelated to Comfort Noise packets generated if Voice Activity Detection is enabled.

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. This feature is enabled by default.

Polycom NoiseBlock™

Polycom NoiseBlock technology automatically mutes the microphone during video calls when a user stops speaking to silence noises that interrupt conversations such as paper shuffling, food wrappers, and keyboard typing. When a user speaks, the microphone is automatically unmuted.

Configuring Polycom NoiseBlock

The following parameters configure the Polycom NoiseBlock feature.

Polycom NoiseBlock Parameters

Parameter Template	Permitted Values
`voice.ns.hf.blocker` new.cfg	1 (default) - Enable Polycom NoiseBlock. 0 - Disable Polycom NoiseBlock.

Audio Output Options

RealPresence Trio 8800 offers audio output and routing options.

By default, audio plays out on the RealPresence Trio 8800 speaker. When you add video capability by connecting and pairing the RealPresence Trio Visual+ system, you can choose to play out audio on connected external speaker and/or the TV/monitor speakers. You can choose audio output options using the parameter `up.audio.networkedDevicePlayout`.

You can configure the following audio routing options:

- RealPresence Trio 8800 speaker only
- Polycom® RealPresence Trio™ Expansion Microphones

  The expansion microphones include a 2.1 m | 7 ft cable that you can attach directly to the RealPresence Trio 8800 to broaden its audio range to a total of 70 ft.
- RealPresence Trio Visual+ using HDMI or a connected 3.5mm analog output
- Any combination of outputs available with RealPresence Trio 8800 and RealPresence Trio Visual+

Use the parameter in the following table to choose where audio is routed to for audio and video calls.

Audio Output Parameters

Parameter Template	Permitted Values
`up.audio.networkedDevicePlayout` new.cfg	`PhoneOnly` (default) —Audio plays out on the RealPresence Trio 8800 speakers. `TvOnly` —Audio plays out on the TV/monitor speakers connected by HDMI to a paired RealPresence Trio Visual+ and, if connected, external speakers connected to the 3.5mm port of a paired RealPresence Trio Visual+. `Auto` —Audio-only calls play out on the RealPresence Trio 8800 speakers. Video-call audio plays out on the TV/monitor speakers connected by HDMI to a paired RealPresence Trio Visual+ and, if connected, external speakers connected to the 3.5mm port of a paired RealPresence Trio Visual+.
`feature.usb.device.hostOs` new.cfg	Specify the operating system of the computer you are connecting by USB when using RealPresence Trio as an audio output device. Windows (default) - The computer connected by USB to the RealPresence Trio uses a Windows operating system. Other - The operating system of the computer connected via USB to the RealPresence Trio system is other than Windows or Mac. Mac - The computer connected by USB to the RealPresence Trio uses a Mac operating system. Confirm - The user is prompted the computer's operating system each time a USB cable is used to connect the RealPresence Trio 8800 system.

USB Calls

You can use RealPresence Trio 8800 as an audio device for your tablet or laptop by connecting your device to the RealPresence Trio 8800 over Bluetooth or with the USB cable supplied in the box with the RealPresence Trio 8800 conference phone.

You can use the RealPresence Trio 8800 system as an audio speakerphone when connected by USB to Mac computers running one of the following software versions:

- OS X 10.9.x (Mavericks)
- OS X 10.10.x (Yosemite)
- OS X 10.11.x (El Capitan)

Configuring USB Calls

You can configure settings for USB calls.

USB Call Parameters

Parameter Template	Permitted Values
`device.baseProfile` device.cfg	Generic - Disables the Skype for Business graphic interface. Lync - Use this Base Profile for Skype for Business deployments. SkypeUSB - Use this Base Profile when you want to connect RealPresence Trio to a Microsoft Room System or a Microsoft Surface Hub.
`voice.usb.holdResume.enable` feature.cfg	0 (default) - The Hold and Resume buttons do not display during USB calls. 1 - The Hold and Resume buttons display during USB calls. This parameter applies only when RealPresence Trio Base Profile is set to 'SkypeUSB'.

Audio Sound Effects

You can customize the audio sound effects that play 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.

Sampled Audio Files

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 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)
- mono 8 kHz G.711 u-Law
- G.711 A-Law
- mono 8 kHz A-law/mu-law
- L8/16000 (16-bit, 8 kHz sampling rate, mono)
- L16/16000 (16-bit, 16 kHz sampling rate, mono)

Configuring Sampled Audio Files

Your custom sampled audio files must be available at the path or URL specified in the parameter `saf.x` so the phone can download the files. Make sure to include the name of the file and the .wav extension in the path.

Use the parameters in the following tables to customize this feature.

In the following table, *x* is the sampled audio file number.

Sample Audio File Parameter

Parameter Template	Permitted Values
`saf.x` site.cfg	Specify a path or URL for the phone to download a custom audio file. • Null— the phone uses a built-in file. • Path Name —During startup, the phone attempts to download the file at the specified path in the provisioning server. • URL— During startup, the phone attempts to download the file from the specified URL on the Internet. Must be a RFC 1738-compliant URL to an HTTP, FTP, or TFTP wave file resource. **Note**: A TFTP URL must be in the following format: `tftp://<host>/[pathname]<filename>`. For example: `tftp://somehost.example.com/sounds/example.wav`. **Note:** 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`.

Default Sample Audio Files

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

Sound Effect Patterns

You can specify the sound effects that play for different phone functions. You can also specify the sound effect patterns and the category.

Configuring Sound Effects

Keep the following in mind when using the parameters in the following table:

- Xis the pattern name.
- Y is the instruction number.
- Both x and y need to be sequential.
- Cat is the sound effect pattern category.

  There are three categories of sound effect patterns that you can use to replace cat in the parameter names: `callProg` (Call Progress Tones), `ringer` (Ringtone Patterns) and `misc` (Miscellaneous Patterns).

Sound Effects Parameters

Parameter Template	Permitted Values
`se.pat.callProg.secondaryDialTone.name` region.cfg	1-255
`se.pat.callProg.secondaryDialTone.inst.1.type` region.cfg	0-255
`se.pat.callProg.secondaryDialTone.inst.1.value` region.cfg	0-50
`se.pat.callProg.secondaryDialTone.inst.1.atten` region.cfg	Sound effects name. UTF-8 encoded string
`se.pat.cat.x.inst.y.type` region.cfg	Type of sound effect. sample chord silence branch
`se.pat.cat.x.inst.y.value` region.cfg	The instruction: `sampled` – sampled audio file number, `chord` – type of sound effect, `silence` – silence duration in ms, `branch` – number of instructions to advance. String

Call Progress Tones

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)
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

Ringtone Patterns

The following table lists the ring pattern names and their default descriptions. Note that sampled audio files 1 to 10 listed in the table all use the same built-in file unless that file has been replaced with a downloaded file.

Ringtone Pattern Names

Parameter Name	Ringtone Name	Description
ringer1	Silent Ring	Silent ring **Note:** Silent ring provides a visual indication of an incoming call, but no audio indication.
ringer2	Low Trill	Long single A3 Db3 major warble

Ringtone Pattern Names

Parameter Name	Ringtone Name	Description
ringer3	Low Double Trill	Short double A3 Db3 major warble
ringer4	Medium Trill	Long single C3 E3 major warble
ringer5	Medium Double Trill	Short double C3 E3 major warble
ringer6	High Trill	Long single warble 1
ringer7	High Double Trill	Short double warble 1
ringer8	Highest Trill	Long single Gb3 A4 major warble
ringer9	Highest Double Trill	Short double Gb3 A4 major warble
ringer10	Beeble	Short double E3 major
ringer11	Triplet	Short triple C3 E3 G3 major ramp
ringer12	Ringback-style	Short double ringback
ringer13	Low Trill Precedence	Long single A3 Db3 major warble Precedence
ringer14	Ring Splash	Splash
ringer15	Ring16	Sampled audio file 1
ringer16	Ring17	Sampled audio file 2
ringer17	Ring18	Sampled audio file 3
ringer18	Ring19	Sampled audio file 4
ringer19	Ring20	Sampled audio file 5
ringer20	Ring21	Sampled audio file 6
ringer21	Ring22	Sampled audio file 7
ringer22	Ring23	Sampled audio file 8
ringer23	Ring24	Sampled audio file 9
ringer24	Ring25	Sampled audio file 10

Miscellaneous Patterns

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

Miscellaneous Pattern Names

negativeConfirm	negative confirmation	Negative confirmation
positiveConfirm	positive confirmation	Positive confirmation
remoteHoldNotification	remote hold notification	Remote hold notification
welcome	welcome	Welcome (boot up)

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, which conserves network bandwidth.

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.

Configure Voice Activity Detection

Use the parameters in the following table to configure this feature.

Voice Activity Detection Parameters

Parameter Template	Permitted Values
`voice.vad.signalAnnexB` site.cfg	0—There is no change to SDP. If `voice.vadEnable` is set to 0, add parameter line `a=fmtp:18 annexb="no"` below the `a=rtpmap…` parameter line (where “18” could be replaced by another payload). 1 (default)—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).

Voice Activity Detection Parameters

Parameter Template	Permitted Values
`voice.vadEnable` site.cfg	0— Voice activity detection (VAD) is disabled. 1— VAD is enabled.
`voice.vadThresh` site.cfg	The threshold for determining what is active voice and what is background noise in dB. integer from 0 - 30 25 (default) 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.

Comfort Noise Payload Packets

When enabled, the Comfort Noise payload type is negotiated in Session Description Protocol (SDP) with the default of 13 for 8 KHz codecs, and a configurable value between 96 and 127 for 16 KHz codecs.

Configuring Comfort Payload Packets

The following table lists the parameters you can use to enable Comfort Noise Control.

Comfort Noise Parameters

Parameter Template	Permitted Values
`voice.CNControl` site.cfg	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. 1 (default)—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. 0—Does not publish support or payloads for Comfort Noise.
`voice.CN16KPayload` site.cfg	Alters the dynamic payload type used for Comfort Noise RTP packets for 16 KHz codecs. 96 to 127 122 (default)

Dual-Tone Multi-Frequency (DTMF) Tones

The phone generates dual-tone multi-frequency (DTMF) tones in response to user dialing on the dialpad. 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.

Configuring DTMF Tones

Use the parameters in the following table to set up this feature.

Dual-Tone Multi-Frequency (DTMF) Tone Generation Parameters

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

DTMF Event RTP Payload

Polycom phones are compatible with RFC 2833—RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals, which 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.

Configuring DTMF Event RTP Payload

Use the parameters in the following table to set up this feature.

DTMF Event RTP Payload Parameters

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

Polycom phones use advanced acoustic echo cancellation (AEC) for handsfree operation using the speakerphone. The phones use both linear and non-linear techniques to aggressively reduce echo while permitting natural, full-duplex communication patterns.

See the table RealPresence Trio Solution Audio Codec Support for a list of audio codecs available for each phone and their priority.

Configuring Acoustic Echo Cancellation

Use the parameters in the following table to set up this feature.

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

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.		

RealPresence Trio Solution Audio Codec Support

The following table details the supported audio codecs and priorities for Polycom phone models.

Note that the Opus codec is not compatible with G.729 and iLBC. If you set Opus to the highest priority, G.729 and iLBC are not published; if you set G.729 and iLBC to the highest priority, Opus is not published.

Audio Codec Priority

Phone	Supported Audio Codecs	Priority
RealPresence Trio 8800	G.711 µ -law	6
	G.711a-law	7
	G.722	4
	G.719 (64kbps)	0
	G.722.1 (32kbps)	5
	G.722.1C (48kbps)	2
	G.729AB	8
	Opus	0
	iLBC (13.33kbps, 15.2kbps)	0,0

Audio Codec Support

The following table summarizes the specifications for 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

Audio Codec Specifications

Algorithm	Reference	Raw Bit Rate	IP Bit Rate	Sample Rate	Default Payload Size	Effective Audio Bandwidth
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
Siren 7	SIREN7	16 Kbps 24 Kbps 32 Kbps	32 Kbps 40 Kbps 48 Kbps	16 Ksps	20 ms	7 KHz
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.

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).

Configuring Audio Codecs

Use the parameter in the following table to specify the priority for audio codecs on your Polycom phones.

Audio Codec Priorities Parameters

Parameter Function	**template** > parameter
To specify the priority for a codec.	**site.cfg** > voice.codecPref.<nameOfCodec>

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. 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.

Configuring IP Type-of-Service

Use the parameters in the following table to configure this feature.

IP Type-of-Service (ToS) Parameters

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.*
Set the IP header bits for RTP video.	**site.cfg** > qos.ip.rtp.video.*
.* indicates grouped parameters. See the section Configuring Phone Groups with the Master Configuration File for more information.	

IEEE 802.1p/Q

The phone tags all Ethernet packets it transmits with an 802.1Q VLAN header when the following occurs:

- A valid VLAN ID is specified in the phone's network configuration.
- The phone is instructed to tag packets through Cisco Discovery Protocol (CDP) running on a connected Ethernet switch.
- A VLAN ID is obtained from DHCP or LLDP (see DHCP Settings)

Configuring IEEE 802.1p/Q

You can set the 802.1p/Q `user_priority` field to a value from 0 to 7, and you can configure `user_priority` specifically for RTP packets and call control packets, such as SIP signaling packets, with default settings configurable for all other packets.

Use the parameter in the following table to set values for this feature.

IEEE 802.1p/Q Parameters

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

Voice Quality Monitoring (VQMon)

You can configure the phones to generate various quality metrics that 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 supports RFC6035 compliance as well as draft implementation for voice quality reporting.

You need a license key to activate the VQMon feature on the VVX 300/301, 310/311, 400/401, and 410/411 business media phones. This feature is available for open SIP environments, but is not available with Skype for Business Server. For more information on VQMon, contact your Certified Polycom Reseller.

Configuring VQMon

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 using the parameters 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 generated 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) Parameters

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 Configuring Phone Groups with the Master Configuration File for more information.	

Call Controls and Phone Alerts

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

Microphone Mute

All phones have a microphone mute button. By default, 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.

You cannot configure the microphone mute feature.

Persistent Microphone Mute

With this feature, you can enable the microphone mute to persist across all calls managed on a phone. By default, users can mute the microphone during an active call, and the microphone is unmuted when the active call ends. With persistent microphone mute enabled, when a user mutes the microphone during an active call, the microphone remains muted for all following calls until the user unmutes the microphone or the phone restarts.

\When a user mutes the microphone when the phone is idle, the mute LED glows but no icon displays on the screen. When a user initiates a new active call with the microphone muted, the mute LED glows and a Mute icon displays on the phone screen.

Configuring Persistent Microphone Mute

Use the following parameter to enable persistent microphone mute.

Persistent Mute Parameters

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

Call Timer

By default, a call timer displays on the phone’s screen during calls, and a separate call duration timer displays the hours, minutes, and seconds for each call in progress.

You cannot configure the display of the call timer.

Called Party Identification

By default, the phone displays and logs the identity of all parties called from the phone. The phone obtains called party identities from network signaling. Because called party identification is a default feature, the phone displays caller IDs matched to the call server and does not match IDs to entries in the contact directory or corporate directory.

Connected Party Identification

By default, the phone displays and logs the identities of remote parties you connect to if the call server can derive the name and ID from network signaling. 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's. 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. The phone does not match party IDs to entries in the contact directory or the corporate directory.

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 enable the phones to display the name assigned to contacts in the contact directory. However, the phone cannot match the identity of calling parties to entries in the corporate directory.

Configuring Calling Party Identification

Use the parameters in the following table to configure this feature.

Calling Party Identification Parameters

Parameter Function	**template** > parameter
Substitute the network address ID with the Contact Directory name.	**features.cfg** > up.useDirectoryNames
Override the default number of calls per line key for a specific line.	**reg-advanced.cfg** > reg.x.callsPerLineKey

SIP Header Warnings

You can configure the warning field from a SIP header to display a pop-up message on the phone, for example, when a call transfer failed due to an invalid extension number. You can display pop-up messages in any language supported by the phone. The messages display for three seconds unless overridden by another message or action. For more information on SIP headers, refer to the section Supported SIP Request Headers.

Configuring SIP Header Warnings

You can use the parameters in the following table to enable the warning display or specify which warnings to display.

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

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.

Configuring Distinctive Call Waiting

Distinctive Call Waiting Parameters

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

Do Not Disturb

You can enable Do Not Disturb (DND) locally on the phone or on the server. The local DND feature is enabled by default, and users can enable or disable DND for all or individual registered lines on the phone. When enabled, users are not notified of incoming calls placed to their line.

Server-Based Do Not Disturb

If you want to enable server-based DND, you must enable the feature on both a registered phone and on the server. The following conditions apply for server-based DND:

- Server-based DND can be applied to multiple registered lines on a phone; however, applying DND to individual registrations is not supported.
- Server-based DND cannot be enabled on a phone configured as a shared line.
- If server-based DND is enabled but not turned on when the DND feature is enabled on the phone, the “Do Not Disturb” message displays on the phone, but incoming calls continue to ring.
- Server-based DND disables local Call Forward and DND, however, if an incoming is not routed through the server, an audio alert still plays on the phone.

Configuring Do Not Disturb

Use the parameters in the following table to configure the local DND feature.

Do Not Disturb Parameters

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

Call Waiting Alerts

By default, the phone alerts users to incoming calls while a user is in an active call. You can choose to disable call waiting alerts and specify ringtones for incoming calls.

Configuring Call Waiting Alerts

Use the parameters in the following table to configure call waiting alerts.

Call Waiting Alerts Parameters

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

Missed Call Notifications

By default, a counter with the number of missed calls displays on the Recent Calls icon on the phone. You can configure the phone to record all missed calls or to display only missed calls that arrive through the SIP server. You can also enable missed call notifications for each registered line on a phone.

Configuring Missed Call Notifications

Use the following table to configure options for the missed call notifications feature.

Missed Call Notification Parameters

Parameter Function	**template** > parameter

Missed Call Notification Parameters

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

Synthesized Call Progress Tones

Polycom phones play call signals and alerts, called call progress tones, that include busy signals, ringback sounds, and call waiting tones. The built-in call progress tones match standard North American tones. If you want to customize the phone's call progress tones to match the standard tones in your region, contact Polycom Support.

Call Hold

Call hold enables users to pause activity on an active call so that they can use the phone for another task, such as searching the phone's menu for information. When an active call is placed on hold, a message displays informing the held party that they are on hold.

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.

Configuring Call Hold

See the following table for a list of available parameters you can configure for this feature.

Call Hold Parameters

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 Configuring Phone Groups with the Master Configuration File for more information.

Call Transfer

The call transfer feature enables users to transfer an existing active call to a third-party address. You can can configure the call transfer feature and set the default transfer type.

Users can perform following 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. 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 ringing 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.

Configuring Call Transfer

Use the following table to specify call transfer behavior.

Call Transfer Parameters

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
Specifies if the Transfer soft key displays while a transferred call is in the ringing state.	`voIpProt.SIP.allowTransferOnProceeding`

Call Forwarding

Polycom phones support a flexible call forwarding feature that enables users to forward incoming calls to another contact or phone line. Users can enable call forwarding in the following ways:

- To all calls
- To incoming calls from a specific caller or extension
- During an incoming call
- When the phone is busy
- When do not disturb is enabled
- After a set number of rings before the call is answered
- To a predefined destination chosen by the user

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

- Forward to a contact
- Forward to voicemail
- Forward to Delegates
- Simultaneously Ring Delegates
- Simultaneously Ring Group Contacts

Call Forward on Shared Lines

You can enable server-based call forwarding for shared lines. If using BroadWorks R20 server, note the following:

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

The server-based and local call forwarding features do not work with the shared call appearance (SCA) and bridged line appearance (BLA) features. In order to enable users to use call forwarding, disable SCA or BLA enabled.

Configuring Call Forwarding

Use the parameters in the following table to configure feature options for call forwarding. No parameters are needed to enable call forwarding on Skype for Business-enabled phones.

Call Forwarding Parameters

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.*
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

Call Forwarding Parameters

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

.* indicates grouped parameters. See the section Configuring Phone Groups with the Master Configuration File for more information.

Automatic Off-Hook Call Placement

You can configure the phone to automatically place a call to a specified number when the phone goes off-hook, which is sometimes referred to as Hot Dialing. The phone goes off-hook when a user lifts the handset, presses the New Call soft key, or presses the speakerphone buttons on the phone.

Configuring Automatic Off-Hook Call Placement

As shown in the following table, you can specify an off-hook call contact, enable or disable the feature for each registration, and specify a protocol for the call.

You can specify only one line registration for the RealPresence Trio 8800 system.

Automatic Off-Hook Call Placement Parameters

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.	**reg-advanced** > call.autoOffHook.x.protocol

Multiple Line Keys Per Registration

You can assign a single registered phone line address to multiple line keys on Polycom phones. This feature can be useful for managing a high volume of calls to a single line. This feature is not supported when registered with Microsoft Skype for Business Server.

Configuring Multiple Line Keys Per Registration

Use the parameter in the following table to configure this feature. This feature is one of several features associated with Call Appearances.

Multiple Line Keys Per Registration Parameters

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

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 user interface. For example, with multiple call appearances, users can place one call on hold, switch to another call on the same registered line, and have both calls display on the phone.

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 Multiple Line Keys Per Registration. If you assign a registered line to multiple line keys, the default number of concurrent calls applies to all line keys.

RealPresence Trio can have a maximum of 12 concurrent calls with only one active call in progress. You can register one line on the RealPresence Trio system.

Configuring Multiple Call Appearances

Use the parameters in the following table to set the maximum number of concurrent calls per registered line and the default number of calls per line key.

Note that you can set the value for the `reg.1.callsPerLineKey` parameter to a value higher than 1, for example, 3. After you set the value to 3, for example, you can have three call appearances on line 1. By default, any additional incoming calls are automatically forwarded to voicemail. If you set more than two call appearances, a call appearance counter displays at the top-right corner on the phone.

Multiple Call Appearances Parameters

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

Shared Call Appearances

Shared call appearance enables an active call to display simultaneously on multiple phones in a group. All call states of a call—active, inactive, on hold—are displayed on all phones of a group.

By default, the answering phone has sole access to the incoming call, which is called line seize. If the answering phone places the call on hold, that call becomes available for pickup to all phones in that group. You can enable other phones in the group the ability to enter a conversation on one of the group phones, which is referred to as a barge in.

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.

Configuring Shared Call Appearances

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.

Use the parameters in the following table to configure options for this feature.

Shared Call Appearances Parameters

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

.* indicates grouped parameters. See the section Configuring Phone Groups with the Master Configuration File for more information.

Private Hold on Shared Lines

Enable the private hold feature to display the PvtHold soft key and enable users to hold calls without notifying other phones registered with the shared line. 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.

Configuring Private Hold on Shared Lines

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.

Use the parameters in the following table to configure this feature.

Private Hold Parameters

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.	**features.cfg** > reg.x.enablePvtHoldSoftKey

Bridged Line Appearance

Bridged line appearance connects calls and lines to multiple phones. 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, which is called 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.

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 only with shared call appearances.

Configuring Bridged Line Appearance

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.

Use the parameters in the following table to configure this feature.

Bridged Line Appearance Parameters

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

Voicemail Integration

When you configure Polycom phones with a SIP URL that integrates with a voicemail server contact, users receive a visual and audio alert when they have new voicemail messages available on their phone.

Configuring Voicemail Integration

You can configure a message waiting alert on the phone to indicate when users have unread voicemail messages.

Use the parameters in the following table to configure this feature.

Voicemail Integration Parameters

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.	**features.cfg** > up.mwiVisible
Specify if the phone screen backlight illuminates when you receive a new voicemail message.	**site.cfg** > mwi.backLight.disable

Local Call Recording

Local call recording enables you to record audio calls to a USB device connected to the phone. You can play back recorded audio on the phone or devices that run applications like Windows Media Player® or iTunes® on a Windows® or Apple® computer. To use this feature, ensure that the USB port is enabled.

Audio calls are recorded in **.wav** format and include a date/time stamp. The phone displays the recording time remaining on the attached USB device, and users can browse all recorded files using the phone's menu.

Federal, state, and/or local laws may legally require that you notify some or all of the call parties when a call recording is in progress.

Configuring Local Call Recording

Use the parameters in the following table to configure local call recording.

Local Call Recording Parameters

Parameter Template	Permitted Values
`feature.callRecording.enabled` features.cfg	0 (default) - Enables audio call recording. 1 - Disables audio call recording.

Local and Centralized Conference Calls on RealPresence Trio

When the RealPresence Trio 8800 system is paired with the RealPresence Trio Visual+ system, users can initiate and join the following types of conferences:

- Local multipoint audio conference with up to four external connections
- Local video conferences
- Video calls on supported H.264 standards-compliant video bridges or services

The RealPresence Trio solution can send and receive one video connection and displays the far-end device that joined the call last. RealPresence Trio does not support locally-hosted multipoint video conferencing.

To enable video and content for conference calls, you must connect RealPresence Trio Visual+ to a monitor and connect a Logitech Webcam C930e USB camera. When the devices are connected and paired, users can send video and share content. For details and limitations of content sharing, refer to the section Content Sharing.

Configuring Local and Centralized Conference Calls

The following table lists available call management parameters.

Local and Centralized Conferences Parameters

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

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.

Creating a Custom Intercom Soft Key

By default, an Intercom soft key displays on the phone, but you have the option to provide users the ability to initiate intercom calls directly to a specified contact using enhanced feature keys (EFKs). 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 for a custom soft key in one of the following ways:

- $FIntercom$

  This is an F type macro that behaves as a custom Intercom soft key. Pressing the soft key opens the Intercom dial prompt users 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 that enables you to specify a Direct intercom button that always calls the number you specify in <number>. No other input is necessary.

Configuring Intercom Calls

Use the parameters in the table to configure the behavior of the calling and answering phone.

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

Group Paging

The group paging feature enables users to make pages —one-way audio announcements—to users subscribed to a page group. There are 25 groups/channels users can subscribe to.

If you are using Group Paging with RealPresence Trio solution, you can only receive incoming pages. You cannot use RealPresence Trio solution to send outgoing pages.

Group paging users can send announcements to recipients subscribed to any of the 25 paging groups. Any announcements sent to the paging group play through the phone's speakerphone.

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.

Configuring Group Paging

Administrators must enable paging before users can subscribe to a page group. You can specify the same IP multicast address in the parameter `ptt.address`. Use the parameters in the following table to configure this feature.

The default port used by Group Paging conflicts with the UDP port 5001 used by Polycom® People+Content™ on the RealPresence Trio system. Since the port used by People+Content is fixed and cannot be configured, configure one of the following workarounds:

- Configure a different port for Group Paging using parameter `ptt.port` or
- Disable People+Content IP using parameter `content.ppcipServer.enabled="0"`.

Group Paging Parameters

Parameter Function	**template** > parameter
Specify the IP multicast address used for the PTT and paging features.	**site.cfg** > ptt.address
Enable paging mode.	**site.cfg** > ptt.pageMode.enable
Specify the display name.	**site.cfg** > ptt.pageMode.displayName
Specify settings for all page groups.	**features.cfg** > ptt.pageMode.group.*
.* indicates grouped parameters. See the section Apply Features and Settings to a Group of Phones for more information.	

Call Logs and Directories

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

Reset Contacts and Recent Calls Lists on RealPresence Trio System

You can reset the Contacts list and Recent call lists are stored locally on the RealPresence Trio 8800 system to their default settings.

To reset the contact and recent calls lists:

1 On the phone, go to **Settings > Advanced.**

2 Enter the administrative password (default 456).

3 Select **Reset to defaults > Reset User Data**.

4 When prompted “Are you sure?”, select **Yes**.

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 on your provisioning server as a file in XML format named ***<MACaddress>*-calls.xml**. If you want to route the call logs to another server, use the `CALL_LISTS_DIRECTORY` field in the master configuration file. All call logs are enabled by default.

The phones automatically maintain the call logs in three separate call lists: Missed Calls, Received Calls, and Placed Calls. Users can clear lists manually on their phones, or you can delete individual records or all records in a group (for example, all missed calls).

Configuring Call Logs

Use the parameters in the following table to configure this feature.

Configure the Call Logs

Parameter Function	**template** > parameter
Enable or disable the missed call list.	**features.cfg** > feature.callListMissed.enabled

Configure the Call Logs

Enable or disable the placed call list.	**features.cfg** > feature.callListPlaced.enabled
Enable or disable the received call list.	**features.cfg** > feature.callListReceived.enabled

Call Log Elements and Attributes

The following table describes each element and attribute that displays in the call log. 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
The line protocol.	
startTime	String
The start time of the call. For example: 2010-01-05T12:38:05 in local time.	
duration	String
The duration of the call, beginning when it is connected and ending when the call is terminated. For example: `PT1H10M59S`.	
count	Positive Integer
The number of consecutive missed and abandoned calls from a call destination.	
destination	Address
The original destination of the call. For outgoing calls, this parameter designates the outgoing call destination; the name is initially supplied by the local phone (from the name field of a local contact entry) but may later be updated via call signaling. This field should be used for basic redial scenarios. For incoming calls, the called destination identifies the requested party, which may be different than any of the parties that are eventually connected (the destination may indicate a SIP URI which is different from any SIP URI assigned to any lines on the phone).	
source	Address
The source of the call (caller ID from the call recipient’s perspective).	

Call Log Elements and Attributes

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.	

Local Contact Directory

Polycom phones feature a contact directory file you can use to store frequently used contacts. The UC Software package includes a template contact directory file named **000000000000-directory~.xml** that 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.

Maximum Capacity of the Local Contact Directory

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 set a lower maximum number of contacts for the phones.

Maximum File Size and Number of Contacts

Phone	Maximum File Size	Maximum Number of Contacts in File
RealPresence Trio 8800	4MB	2000

Creating Per-Phone Directory Files

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.

Maintaining Per-Phone Directory Files

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.

Configuring the Local Contact Directory

The following parameters configure the local contact directory.

Local Contact Directory Parameters

Parameter Template	Permitted Values
`dir.local.contacts.maxNum` features.cfg	Set the maximum number of contacts that can be stored in the Local Contact Directory. The maximum number varies by phone model, refer to section ‘Maximum Capacity of the Local Contact Directory’. 1 - 3000
`dir.local.readOnly` features.cfg	0 (default) - Disable read only protection of the local Contact Directory. 1 - Enable read-only protection of the local Contact Directory.
`feature.directory.enabled` features.cfg	0 (default) - The local contact directory is disabled when the RealPresence Trio solution Base Profile is set to Lync. 1 - The local directory is enabled when the RealPresence Trio solution Base Profile is set to Lync.

Local Contact Directory Parameters

Parameter Template	Permitted Values
`dir.search.field` features.cfg	Specify whether to search the directory by first name or last name. 0 (default) - Contact directory searches are sorted by contact's last name. 1 - Contact directory searches are sorted by first name.
`voIpProt.SIP.specialEvent.checkSync.downloadDirectory` site.cfg	Specify whether the phone downloads the updated global directory file after receiving a check-sync NOTIFY message. 0 (default) - The phone only downloads software and configuration updates after receiving a `checksync NOTIFY` message. 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.

Local Contact Directory Parameter Elements

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

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

Speed Dials

Local Contact Directory Parameter Elements

Element	Definition	Permitted Values
fn	First Name	UTF-8 encoded string of up to 40 bytes[1]
The contact's first name.		
ln	Last Name	UTF-8 encoded string of up to 40 bytes[1]
The contact's last name.		
ct	Contact	UTF-8 encoded string containing digits (the user part of a SIP URL) or a string that constitutes a valid SIP URL
Used by the phone to address a remote party in the same way that a string of digits or a SIP URL are dialed manually by the user. This element is also used to associate incoming callers with a particular directory entry. The maximum field length is 128 characters. Note: This field cannot be null or duplicated.		
sd	Speed Dial Index	VVX=Null, 1 to 9999 RealPresence Trio=20
Associates a particular entry with a speed dial key for one-touch dialing or dialing.		
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.		
pt	Protocol	SIP 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.		
bw	Buddy Watching	0 or 1

Local Contact Directory Parameter Elements

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.

You can link entries in the local contact directory to speed dial contacts to line keys on the Home or Lines screen to enable users to place calls quickly using dedicated speed dial buttons. To set up speed dial through the phone’s contact directory, see the section Local Contact Directory.

Speed Dial Index Range

You can assign contacts as speed dials using the speed dial index ranges listed in the following table.

Speed Dial Index Ranges

Phone Model	Range
RealPresence Trio 8800	1 - 20

Configuring Speed Dials

After setting up you per-phone directory file (***<MACaddress>*-directory.xml),** 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. See Local Contact Directory.

Note that 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.

Use the parameters in the following table, which identifies the directory XML file and the parameters you need to set up your speed dial contacts.

Configure the Speed Dial Feature

Parameter Function	**template** > parameter
Configure the maximum number of speed dial contacts that can display on the RealPresence Trio Home screen.	`dir.local.contacts.maxFavIx`
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

Corporate Directory

You can connect phones to a corporate directory server that supports the Lightweight Directory Access Protocol (LDAP), version 3. After you set up the corporate directory on the phones, users can search for contacts in the directory, place calls to directory contacts, and save entries 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 also 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.

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*.

Configuring the Corporate Directory

Use the parameters in the following table to configure this feature. Note that the exact configuration of a corporate directory depends on the LDAP server you use.

For detailed explanations and examples of all currently supported LDAP directories, see *Technical Bulletin 41137*: *Best Practices When Using Corporate Directory on Polycom Phones* 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 Apply Features and Settings to a Group of Phones for more information.	

Local Digit Map

The phone has a local digit map feature that, when configured, automatically calls a dialed number, which eliminates the need for a user 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 dialed number matches any string of a digit map, the call is automatically placed. If a dialed number 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 configuration syntax of the digit map is based on recommendations in section 2.1.5 of RFC 3435.

Configuring Local Digit Maps

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

Use the parameters in the following table to configure this feature.

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.*
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 Apply Features and Settings to a Group of Phones for more information.	

Open SIP Digit Map

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 uses to find the shortest possible match.

The following is a list of digit map string rules for open SIP environments.

- 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.
- 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, ***R****555****R****604****R*** would replace 555 with 604. Digit map timer letter T indicates a timer expiry. Digit map protocol letters *S* and *H* indicate the protocol to use when placing a call.

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

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*.

Generating Secondary Dial Tone with Digit Maps

You can regenerate a dial tone by adding a comma "**,**" to the digit map. You can dial seven-digit numbers after dialing "8" as shown next in the example 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 the digit "8", the dial tone plays again, and users can complete the remaining seven-digit number. In this example, if users also have a 4-digit extension that begins with "8", then users 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
```

Hardware and Accessories

This section provides information on configuring phone hardware.

Powering the RealPresence Trio 8800 Solution

You can power the RealPresence Trio 8800 with Power over Ethernet (PoE) or PoE+ (IEEE 802.3at Type 2). When RealPresence Trio 8800 is booting up, an on-screen message indicates the available power supply type. Note that PoE+ provides RealPresence Trio 8800 with full functionality.

The following features are not available on RealPresence Trio 8800 when using PoE:

- The RealPresence Trio 8800 LAN OUT port out does not provide PoE+ power and cannot be used to power the RealPresence Trio Visual+.
- No USB charging is provided to devices (mobile phones, tablets) connected to the RealPresence Trio 8800 USB port.
- Maximum peak power to the loudspeaker is limited.

Power the RealPresence Trio 8800 System with the Optional Power Injector

If your building is not equipped with PoE+ you can use the optional power injector to provide PoE+ and full functionality to RealPresence Trio 8800.

When using the power injector to power the RealPresence Trio 8800, you must connect cables in the following sequence:

1. Plug the AC power cord of the power injector into the wall and use a network cable to connect the power injector to the RealPresence Trio 8800.
2. Connect the power injector to the network with a CAT-5E or CAT-6 Ethernet cable.

The power adapter LED is green when the RealPresence Trio 8800 is correctly powered. If the LED is yellow, the power injector is bypassed and the RealPresence Trio system is drawing PoE power from the outlet.

If the RealPresence Trio Visual+ loses power after a RealPresence Trio 8800 reboot, unplug both devices and repeat steps 1 and 2.

If the power injector LED is yellow, turn off the PoE network port or connect the RealPresence Trio solution in the following sequence:

1. Power up RealPresence Trio 8800 and Visual+ using the power injector but do not plug the devices into the network wall port.
2. Wait for the RealPresence Trio 8800 and Visual+ systems to boot up
3. Plug the devices into the network wall port.
4. Ensure the LED indicator on the power injector is green.

Powering the RealPresence Trio Visual+ Solution

How you power the RealPresence Trio Visual+ can depend on the power options your building is equipped with. Consider the following setup points:

- If you are using PoE+ or the optional power injector, you can power the RealPresence Trio Visual+ directly from the RealPresence Trio 8800 using an Ethernet cable. In this scenario, you do not need to pair the RealPresence Trio 8800 with the RealPresence Trio Visual+.
- If you are using PoE, you must power the RealPresence Trio Visual+ separately using an Ethernet cable or use the optional power injector. In this scenario, you must pair the RealPresence Trio 8800 with the RealPresence Trio Visual+. For pairing instructions, refer to PAIR.
- If you use PoE+, you have the option to power the RealPresence Trio 8800 and RealPresence Trio Visual+ separately and pair. When powering separately, you do not need to connect the RealPresence Trio 8800 directly to RealPresence Trio Visual+.

Pairing the RealPresence Trio Visual+ with RealPresence Trio 8800

Pair the RealPresence Trio Visual+ with RealPresence Trio 8800 to enable users to place video to calls and share content. You can pair only one RealPresence Trio Visual+ to a RealPresence Trio 8800 system. Polycom recommends you plug both devices into a local gigabit switch.

You can pair the RealPresence Trio Visual+ to the system using configuration files or from the RealPresence Trio 8800 menu system. To pair, the RealPresence Trio 8800 and RealPresence Trio Visual+ must be connected to the same subnet and you must unblock the following network components:

- Multicast address 224.0.0.200
- Port 2000

You cannot use RealPresence Trio Visual+ for video calls when you connect RealPresence Trio 8800 to your network using Wi-Fi. The RealPresence Trio 8800 and RealPresence Trio Visual+ only pair when the RealPresence Trio 8800 is connected to your network over Ethernet.

Pair the RealPresence Trio Solution Manually

You can manually pair the RealPresence Visual+ to the system at any time from the RealPresence Trio 8800 menu.

To pair RealPresence Trio Visual+ with Trio 8800 manually:

1. Set up RealPresence Trio Visual+. For instructions, refer to the RealPresence Trio 8800 Setup Sheet that comes in the packaging box.

   The Welcome screen displays on your monitor and indicates steps to pair with RealPresence Trio 8800.
2. Tap the **Pair** button on RealPresence Trio Visual+ to broadcast discovery to the RealPresence Trio 8800.
3. On the RealPresence Trio 8800, go to **Settings > Advanced > Networked Devices**, and ensure that **Notification of New Devices** is **On**.
4. Choose one of the following:
   - If you have not paired the device before, tap **Pair with New Device**, tap the device you want to pair from the Discovered Devices list, and in the Details screen, tap **Pair**. (All currently paired devices display under Paired Devices.)
   - If the device has been paired before, select the device from the **Available Devices** list and tap **Pair**.
5. When you see the message prompting you to complete pairing, do one of the following:
   - Tap **Complete**.
   - Tap the **Pair** button on the RealPresence Visual+

   If pairing was successful, a success message displays on the monitor along with a self-view window, the LED light on the RealPresence Trio Visual+ device is continuously green, and a paired icon displays on the phone.

   If pairing was not successful, a message displays on the monitor that the devices could not pair.

   After successful pairing, if devices become disconnected for 60 seconds, a message displays that the devices have temporarily lost connection.

Pairing the RealPresence Trio Solution using Configuration Parameters

To pair using configuration files, enter the MAC address of your Visual+ device as the value for the parameter `mr.pair.uid.1`. The MAC address can be in either of the following formats:

- `00e0d::B09128D`
- `00E0DB09128D`.

Use the following parameters to configure this feature and additional feature options.

Pairing Parameters

Parameter Function	**template** > parameter
Enables users with RealPresence Desktop on a laptop or RealPresence Mobile on a tablet to pair with the RealPresence Trio conference phone using SmartPairing.	smartPairing.mode
The relative volume to use for the SmartPairing ultrasonic beacon.	smartPairing.volume
Use SRTP to encrypt and authenticate audio signals.	mr.audio.srtp.require

Pairing Parameters (continued)

Cycles through the background images	mr.bg.selection
Enter the MAC address of the RealPresence Trio Visual+ you want to pair with.	mr.pair.uid.1
Use SRTP to encrypt and authenticate audio media.	mr.audio.srtp.require
Use TLS to communicate between RealPresence Trio 8800 and RealPresence Trio Visual+ systems.	mr.pair.tls.enabled
Enable the camera’s automatic focus.	mr.video.camera.focus.auto
Specify the distance to the camera’s optimally-focused target.	mr.video.camera.focus.range
Choose the minimum time in seconds between transmitted video i-Frames or transmitted i-Frame requests.	mr.video.iFrame.minPeriod

Identify Paired Devices

If you are using multiple RealPresence Trio 8800 systems and are not sure which RealPresence Trio Visual+ it is paired with which, you can identify which devices are paired with the system on the RealPresence Trio 8800.

To identify paired devices:

1 On the phone, go to **Settings > Advanced > Networked Devices**, and ensure that **Notification of New Devices** is **On**.

2 Select a device that displays under Paired Devices or Available Devices.

3 Tap **Identify** to flash the LED of the device you selected.

Place the RealPresence Trio Visual+ in Pairing Diagnostic Mode

If you are using multiple RealPresence Trio 8800 systems and are not sure which RealPresence Trio Visual+ it is paired with which, you can i place the RealPresence Trio Visual+ devices in pairing diagnostic mode to distinguish between accessories.

To enter pairing diagnostic mode:

1 Power up the RealPresence Trio Visual+ device.

2 Wait for the initial LED on state to turn off.

3 Press and hold the pairing button until the LED turns orange.

4 Release the pairing button.

5 The LED blinks.

6 Wait for the device to reboot.

7 The paired Pod LED is steady green.

Phone Interface Lockdown

To increase security, you can disable Wi-Fi, Bluetooth and NFC, and the USB host port and device port.

Configure Phone Interface Lockdown

Use the following parameters to lock phone ports and features.

Phone Interface Lockdown Parameters

Parameter Function	**template** > parameter
Enable or disable the host USB port.	feature.usb.host.enable
Enable or disable the device USB port.	feature.usb.device.enabled
Enable or disable the Bluetooth feature and menu on the phone.	feature.bluetooth.enabled
Enable or disable NFC for Bluetooth pairing.	feature.nfc.enabled
Enable or disable use of Wi-Fi calling.	device.wifi.enabled

Power-Saving

The VVX 500, 600, and 1500 phones and the RealPresence Trio 8800 support a power-saving feature that has a number of options you can configure:

- Turn on the phone's power-saving feature during non-working hours and working hours.

  If you want to turn on power-saving during non-working hours, you can configure the power-saving feature around your work schedule.
- On the VVX 1500, 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.

In an unused conference room where the phone is in idle mode and the display is off, the RealPresence Trio solution has the capability to wake up when a user enters the room, dependent upon the lighting in the room.

When you enable power-saving mode on VVX 500 and 600, the phone display screen does not automatically turn back on after going idle.

Configuring Power-Saving

Use the parameters in the following table to configure the power-saving features and feature options.

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.*
`powerSaving.tvStandbyMode` new.cfg	black (default) - Power-saving mode shows a black screen on the RealPresence Trio Visual+ display. noSignal - Power-saving mode turns off the HDMI signal going to the RealPresence Trio Visual+ display.

.* indicates grouped parameters. See the section Configuring Phone Groups with the Master Configuration File for more information.

Consumer Electronics Controls (CEC) over HDMI

Consumer Electronics Control (CEC) enables system standby on RealPresence Trio systems when using the RealPresence Trio Visual+ system to connect to CEC-capable monitors with HDMI. Check the feature settings and sub-settings on your monitor to verify that your monitor supports CEC.

When you enable CEC, any connected CEC-capable monitors switch to standby mode to save power when the RealPresence Trio system enters standby mode. When the system awakes, the monitors are powered up before displaying RealPresence Trio system video.

CEC features can vary by the brand of monitor. Specifically, some monitors have sub-feature settings under the main CEC setting that control whether or not the monitor responds to CEC commands. Ensure that you enable all CEC features and sub-features on all monitors connected to the RealPresence Trio systems.

Configuring CEC over HDMI

CEC is disabled by default, and you can enable CEC using the Web Configuration Utility or centralized provisioning.

Configure Consumer Electronics Control (CEC) using the Web Configuration Utility

You can enable or disable CEC on RealPresence Trio systems using the Web Configuration Utility.

To enable CEC using the Web Configuration Utility:

1. Enter the IP address of the RealPresence Trio system you are using to a web browser.
2. Log into the Web Configuration Utility as an administrator.

3 Go to **Settings > Networked Devices > Power Saving Settings**.

4 Beside **Consumer Electronic Control**, select **Enable** or **Disable**.

Consumer Electronics Controls (CEC) over HDMI Parameters

Use the parameters in the following table to configure CEC over HDMI for the RealPresence Trio solution.

CEC Power-Saving Parameters

Parameter Template	Permitted Values
`powerSaving.cecEnable` new.cfg	0 (default) - The RealPresence Trio Visual+ display behavior is controlled only by the value set for `powerSaving.tvStandbyMode`. 1 - When the RealPresence Trio 8800 enters power-saving mode, the RealPresence Trio Visual+ display switches to standby mode and powers up when the RealPresence Trio 8800 exits power-saving mode.

Phone Functions

This section provides information on setting up display and appearance features as well as setting up a number of custom phone functions available on the phone.

Bluetooth and Near Field Communication (NFC)

When you enable Bluetooth, users can connect a Bluetooth-enabled device, such as a laptop or mobile phone, to the RealPresence Trio 8800 phone and play audio from audio calls, video calls, music or video players from the conference phone's loudspeaker.

When a device is connected over Bluetooth during an audio or a video call, users can use the conference phone's microphones for audio instead of the microphone(s) of your connected device.

The RealPresence Trio 8800 conference phone can remember up to 10 previously paired devices. Note that users cannot connect via Bluetooth during an active call.

NFC-Assisted Bluetooth Connection

NFC (near field communication)-assisted Bluetooth pairing is disabled by default. When Bluetooth is enabled, you can connect one mobile phone or tablet at a time, place calls on their mobile phone, and use the conference phone as a speaker and microphone for the call. T

Enable or Disable NFC Mode from the Phone Menu

When NFC is enabled and paired the RealPresence Trio 8800 with a device, the NFC logo displays on the screen and users can use the phone to play audio from media, such as music or videos, from their mobile phone.

To enable/disable NFC Mode from the phone:

» Go to **Settings > Advanced > Administrator Settings > NFC Mode**.

Configuring Bluetooth and NFC Mode

Use the parameters in the following table to configure Bluetooth and NFC Mode on the RealPresence Trio 8800 system.

When NFC is enabled and a device is paired to the RealPresence Trio 8800, the NFC logo displays on the screen and users can use the phone to play audio from media, such as music or videos, from their mobile phone.

Configure Bluetooth and NFC Parameters

Parameter Function	**template** > parameter
Enable or disable Bluetooth connection.	**features.cfg** > feature.bluetooth.enabled
Turn the Bluetooth radio (transmitter/receiver) on or off.	**features.cfg** > bluetooth.radioOn
Name the device to which you are connecting over Wi-Fi.	**new.cfg**, **sip-interop.cfg** > bluetooth.devName
Set the Bluetooth discoverable device visibility timeout in seconds.	**new.cfg**, **features.cfg** > bluetooth.discoverableTimeout
Enable or disable NFC-capable devices to the RealPresence Trio 8800 solution.	feature.nfc.enabled

Phone Display and Appearances

This section provides information on setting up features involving the phone’s user interface.

RealPresence Trio User Interface

This section provides a quick overview of icons and feature buttons you can display or hide on the RealPresence Trio 8800 system user interface. You can also configure system information to display on the monitor connected to the RealPresence Visual+ system.

For more information about each option, refer to the section for that feature or search for a parameter.

Configuring the RealPresence Trio User Interface

The following table lists parameters you can use to hide or display icons and features.

Phone Menu	Configuration Parameter	Permitted Values
Bluetooth	`feature.bluetooth.enabled`	1 (default) - Bluetooth connection is enabled and the Bluetooth menu displays. 0 - Bluetooth connection is disabled.
Call Lists	`feature.callList.enabled`	1 (default) - Allows you to enable the missed, placed, and received call lists on all phone menus including the Home screen and dial pad. 0 - Disables all call lists. Hiding call lists from the Home screen and dial pad requires UCS 5.4.2 RevAA or higher.
Missed Calls	`feature.callListMissed.enabled`	1 (default) - Missed calls show in the Missed Calls call list. 0 - Missed calls do not show in the Missed Calls list and you cannot clear existing entries.
Placed Calls	`feature.callListPlaced.enabled`	1 (default) - Placed calls show in the Placed Calls call list. 0 - Placed calls do not show in the Placed Calls list and you cannot clear existing entries.

Phone Menu	Configuration Parameter	Permitted Values
Received Calls	`feature.callListReceived.enabled`	1 (default) - Received calls show in the Received Calls call list. 0 - Received calls do not show in the Received Calls list and you cannot clear existing entries.
Contacts	`feature.contacts.enabled`	1 (default) - Enable display of the Contacts icon displays on the Home screen, the global menu, and in the dialer. 0 - 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.
Global Address Book	`feature.corporateDirectory.alt.enabled`	0 (disable) - The global address book service is disabled. 1 - The global address book service is disabled.
Corporate Directory	`feature.corporateDirectory.enabled`	0 (default) - The corporate directory feature is disabled and the icon is hidden. 1 (default) - The corporate directory is enabled and the icon shows.
Calendar	`feature.exchangeCalendar.enabled`	1 (default) - The calendaring feature is enabled. 0 - The calendaring feature is disabled. You must enable this parameter if you also enable `feature.exchangeCallLog.enabled`. If you disable `feature.exchangeCalendar.enabled`, also disable `feature.exchangeCallLog.enabled` to ensure call log functionality.

Outlook Contacts	`feature.exchangeContacts.enabled`
	`feature.lync.abs.enabled`
		The Outlook Search feature allows you to search and view Outlook Contacts and displays in the Contacts menu when the parameters are set as follows:
		`feature.exchangeContacts.enabled="1"`
		`feature.lync.abs.enabled="0"`

Calendar	`homeScreen.calendar.enable`	1 (default) - The Calendar icon on the Home screen displays. 0 - The Calendar icon does not display on the Home screen and is accessible from the dial pad.

Phone Menu	Configuration Parameter	Permitted Values
Diagnostics	`homeScreen.diagnostics.enable` features.cfg	0 (default) - A Diagnostics icon does not show on the Home screen. 1 - A Diagnostics icon shows on the Home screen to provide quick access to the Diagnostics menu.
Contacts	`homeScreen.directories.enable`	1 (default) - Enable display of the Directories menu icon on the phone Home screen. 0 - Enable display of the Directories menu icon on the phone Home screen.
Settings	`homeScreen.settings.enable`	1 (default) - The Settings menu icon displays on the Home screen and global menu. 0 - The Settings menu icon does not display on the Home screen and global menu. You require UC Software 5.4.2 RevAA or higher to hide the Settings icon from the global menu
Basic Settings	`up.basicSettingsPasswordEnabled`	0 (default) - No password is required to access the Basic settings menu. 1 - A password is required to access the Basic settings menu.
Date and Time	`up.localClockEnabled`	1 (default) - The date and time display. 0 - The date and time do not display.
Voice Mail	`up.oneTouchVoiceMail`	0 (default) - The phone displays a summary page with message counts. Press the Connect soft key to dial the voicemail server. 1 - The phone dials voicemail services directly, if available on the call server, and does not display the voicemail summary page.

Locking the Basic Settings Menu

By default, all users can access the Basic settings menu available on the RealPresence Trio 8800 system and VVX phones. From this menu, users can customize non-administrative features on their phone. You can choose to lock the Basic settings menu to 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.

Configuring the Basic Settings Menu Lock

Use the parameter in the following table to lock the Basic settings menu.

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

Configuring the Phone Theme

You can set the RealPresence Trio solution theme, the labels and colors that display on the user interface.

When the RealPresence Trio's Base Profile is set to Lync, the Skype for Business theme displays by default.

Phone theme Parameters

Parameter Template	Permitted Values
`up.uiTheme` features.cfg	Default (default) - The phone displays the default Polycom theme. SkypeForBusiness - The phone displays the Skype for Business theme.

Configure the RealPresence Trio Monitor Display Information

You can configure the monitor connected to the RealPresence Trio Visual+ to display the system's name, IP address, and extension.

To configure display of system information:

1. Log into the RealPresence Trio's Web Configuration Utility as an Administrator.
2. Configure the settings listed in the following table.

Field Name	Description
System Name	The system name displays at the top left corner of the monitor, and at the top of the Global menu of the RealPresence Trio 8800 system. Specify a system name with the `system.name` parameter, or in the Web Configuration Utility at **Simple Setup > System Name**. Enter a system name that helps user identify the system, for example, '*Conference Room*' or '*Joe's Phone*'. If the `system.name` parameter is not specified, the system name is specified as follows: • If the phone has a registered line: The line label specified by `reg.1.label` is used first as the system name, and if not specified, the phone uses `reg.1.displayName` or `reg.1.address`. • If the phone does not have a registered line: The system name displays as '*RealPresence Trio 8800 (xxxxxx)*' where (*xxxxxx*) is the last six digits of the phone's MAC address.
IP Address	The RealPresence Trio 8800 IP address displays at bottom left of the monitor. You can configure a static IP address in the Web Configuration Utility at **Settings > Network > Ethernet**.
Extension	The extension displays at the bottom center of the monitor. Extension displays the registered line number of the RealPresence Trio 8800. The monitor does not display an extension until the phone registers with a line. For all registered lines (except Microsoft), configure the extension in the Web Configuration Utility at **Simple Setup > SIP Line Identification > Address**. For lines registered with Microsoft, you must configure the extension on the Microsoft server you are using.

Show the RealPresence Trio System IP Address

You can show or hide the IP addresses of the RealPresence Trio 8800 and Visual+ systems.

Parameter Template	Permitted Values
`up.hideSystemIpAddress` features.cfg	Specify where the IP address of the RealPresence Trio 8800 and Visual+ are hidden from view. • Nowhere (default) - The IP addresses display on all user interfaces. • TV - IP addresses are hidden from the TV monitor. • HomeScreen - IP addresses are hidden from the TV monitor and phone menu. • Menus - IP addresses are hidden from the TV monitor, phone Home screen, and menu. • Everywhere - IP addresses are hidden from the TV monitor, phone Home screen, and menu.

Configuring a Status Message

You can choose to display a maximum of five multi-line messages in the RealPresence Trio Visual+ Status Bar. Each message can contain a maximum of 64 characters. If the length of the message exceeds the size of the status bar, the message wraps into multiple lines.

When you configure multiple messages, you can adjust the number of seconds each message displays.

Status Message Parameters

Parameter Template	Permitted Values
`up.status.message.flash.rate` features.cfg	Specify the number of seconds to display a message before moving to the next message. 2 seconds (default) 1 - 8 seconds

`up.status.message.1`
`up.status.message.2`
`up.status.message.3`
`up.status.message.4`
`up.status.message.5`
	<message line one>
	<message line two>
	<message line three>
	<message line four>
	<message line five>

Calendar and Meeting Settings

You can configure several calendar and meeting features.

Meeting Reminder Messages

A meeting reminder displays on the RealPresence Trio solution at five minutes and one minute before the start of a meeting. The five minute reminder disappears after 30 seconds if not dismissed. If the one-minute reminder has not been dismissed, the reminder message displays on the RealPresence Trio 8800 system Home Screen during the duration of the meeting. The one minute reminder disappears when the meeting ends or when the next meeting reminder pops up, whichever comes first.

When multiple meetings are booked at the same time or overlap, a message displays available meetings. Users can tap the message to display the calendar day view and choose which meeting to join.

Configuring Calendar and Meeting Settings

The following parameters configure calendar and meetings settings.

Calendar and Meeting Parameters

Parameter Template	Permitted Values
exchange.meeting.alert.followOfficeHours applications.cfg	1 (default) - Audible alerts occur during business hours. 0 - Audible alerts occur at all times.
exchange.meeting.alertTonePattern applications.cfg	Set the tone pattern of the reminder alerts using Any tone specified by `se.pat.*`. For details, refer to section Customize Audio Sound Effect. positiveConfirm (default)
exchange.meeting.alertToneVolume applications.cfg	Set the volume level of reminder alert tones. 10 (default) 1 - 17
exchange.meeting.hideAllDayNotification applications.cfg	0 (default) - All day meeting notifications display on the Calendar screen. 1 - All day meeting notifications are hidden from the Calendar screen.
exchange.meeting.parseOption1 applications.cfg	Indicates the field in the meeting invite from which the VMR or meeting number should be fetched. Location (default) All LocationAndSubject Description
exchange.meeting.phonePattern applications.cfg	The pattern used to identify phone numbers in meeting descriptions, for example, where "x" denotes any digit and "|" separates alternative patterns: xxx-xxx-xxxx|604.xxx.xxxx. NULL (default0
exchange.meeting.private.promptForPIN applications.cfg	0 (default) - The phone does not prompt users to enter a Skype for Business Conference ID in order to join meetings marked as 'private'. 1 - The phone prompts users to enter a Skype for Business Conference ID in order to join meetings marked as 'private'.
exchange.meeting.reminderEnabled applications.cfg	1 (default), meeting reminders are enabled. 0 - Meeting reminders are disabled.
exchange.meeting.reminderInterval applications.cfg	Set the interval in seconds at which phones display reminder messages. 300 (default) seconds 60 - 900 seconds

Calendar and Meeting Parameters

Parameter Template	Permitted Values
`exchange.meeting.reminderSound.enabled` applications.cfg	1 (default) - The phone makes an alert sound when users receive reminder notifications of calendar events. 0 - The phone does not make an alert sound when users receive reminder notifications of calendar events. Note: When enabled, alert sounds take effect only if `exchange.meeting.reminderEnabled` is also enabled.
`exchange.meeting.reminderType` applications.cfg	Customize the calendar reminder and tone. 2 (default) - Reminders are always audible and visual. 1 - The first reminder is audible and visual reminders are silent. 0 - All reminders are silent.
`exchange.meeting.showOnlyCurrentOrNext`	0 (default) - Disabled the limitation to display only the current or next meeting on the Calendar. 1 - Enables the limitation to display only the current or next meeting on the Calendar.
`exchange.meeting.showTomorrow`	1 (default) - Show meetings scheduled for tomorrow as well as meetings scheduled for today. 0 - Do not show meetings scheduled for tomorrow.
`exchange.menu.location`	Features (default) - Displays the Calendar in the global menu under Settings > Features. Administrator - Displays the Calendar in the Admin menu under Settings > Advanced > Administration Settings.

[1] Change causes phone to restart or reboot.

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. The default value is NULL.

Time Zone Location Parameters

Permitted Values		Permitted Values	
0	(GMT -12:00) Eniwetok,Kwajalein	61	(GMT +2:00) Helsinki,Kyiv
1	(GMT -11:00) Midway Island	62	(GMT +2:00) Riga,Sofia
2	(GMT -10:00) Hawaii	63	(GMT +2:00) Tallinn,Vilnius
3	(GMT -9:00) Alaska	64	(GMT +2:00) Athens,Istanbul
4	(GMT -8:00) Pacific Time (US & Canada)	65	(GMT +2:00) Damascus
5	(GMT -8:00) Baja California	66	(GMT +2:00) E.Europe
6	(GMT -7:00) Mountain Time (US & Canada)	67	(GMT +2:00) Harare,Pretoria
7	(GMT -7:00) Chihuahua,La Paz	68	(GMT +2:00) Jerusalem
8	(GMT -7:00) Mazatlan	69	(GMT +2:00) Kaliningrad (RTZ 1)
9	(GMT -7:00) Arizona	70	(GMT +2:00) Tripoli
10	(GMT -6:00) Central Time (US & Canada)		
11	(GMT -6:00) Mexico City	71	(GMT +3:00) Moscow
12	(GMT -6:00) Saskatchewan	72	(GMT +3:00) St.Petersburg
13	(GMT -6:00) Guadalajara	73	(GMT +3:00) Volgograd (RTZ 2)
14	(GMT -6:00) Monterrey	74	(GMT +3:00) Kuwait,Riyadh
15	(GMT -6:00) Central America	75	(GMT +3:00) Nairobi
16	(GMT -5:00) Eastern Time (US & Canada)	78	(GMT +3:00) Baghdad
17	(GMT -5:00) Indiana (East)	76	(GMT +3:00) Minsk
18	(GMT -5:00) Bogota,Lima	77	(GMT +3:30) Tehran
19	(GMT -5:00) Quito	79	(GMT +4:00) Abu Dhabi,Muscat
20	(GMT -4:30) Caracas	80	(GMT +4:00) Baku,Tbilisi
21	(GMT -4:00) Atlantic Time (Canada)	81	(GMT +4:00) Izhevsk,Samara (RTZ 3)
22	(GMT -4:00) San Juan	82	(GMT +4:00) Port Louis
23	(GMT -4:00) Manaus,La Paz	83	(GMT +4:00) Yerevan
24	(GMT -4:00) Asuncion,Cuiaba	84	(GMT +4:30) Kabul
25	(GMT -4:00) Georgetown	85	(GMT +5:00) Ekaterinburg (RTZ 4)
26	(GMT -3:30) Newfoundland	86	(GMT +5:00) Islamabad
27	(GMT -3:00) Brasilia	87	(GMT +5:00) Karachi
28	(GMT -3:00) Buenos Aires	88	(GMT +5:00) Tashkent
29	(GMT -3:00) Greenland	89	(GMT +5:30) Mumbai,Chennai
30	(GMT -3:00) Cayenne,Fortaleza	90	(GMT +5:30) Kolkata,New Delhi

Permitted Values		Permitted Values	
31	(GMT -3:00) Montevideo	91	(GMT +5:30) Sri Jayawardenepura
32	(GMT -3:00) Salvador	92	(GMT +5:45) Kathmandu
33	(GMT -3:00) Santiago	93	(GMT +6:00) Astana,Dhaka
34	(GMT -2:00) Mid-Atlantic	94	(GMT +6:00) Almaty
35	(GMT -1:00) Azores	95	(GMT +6:00) Novosibirsk (RTZ 5)
36	(GMT -1:00) Cape Verde Islands	96	(GMT +6:30) Yangon (Rangoon)
37	(GMT 0:00) Western Europe Time	97	(GMT +7:00) Bangkok,Hanoi
38	(GMT 0:00) London,Lisbon	98	(GMT +7:00) Jakarta
39	(GMT 0:00) Casablanca	99	(GMT +7:00) Krasnoyarsk (RTZ 6)
40	(GMT 0:00) Dublin	100	(GMT +8:00) Beijing,Chongqing
41	(GMT 0:00) Edinburgh	101	(GMT +8:00) Hong Kong,Urumqi
42	(GMT 0:00) Monrovia	102	(GMT +8:00) Kuala Lumpur
43	(GMT 0:00) Reykjavik	103	(GMT +8:00) Singapore
44	(GMT +1:00) Belgrade	104	(GMT +8:00) Taipei,Perth
45	(GMT +1:00) Bratislava	105	(GMT +8:00) Irkutsk (RTZ 7)
46	(GMT +1:00) Budapest	106	(GMT +8:00) Ulaanbaatar
47	(GMT +1:00) Ljubljana	107	(GMT +9:00) Tokyo,Seoul,Osaka
48	(GMT +1:00) Prague	108	(GMT +9:00) Sapporo,Yakutsk (RTZ 8)
49	(GMT +1:00) Sarajevo,Skopje	109	(GMT +9:30) Adelaide,Darwin
50	(GMT +1:00) Warsaw,Zagreb	110	(GMT +10:00) Canberra
51	(GMT +1:00) Brussels	111	(GMT +10:00) Magadan (RTZ 9)
52	(GMT +1:00) Copenhagen	112	(GMT +10:00) Melbourne
53	(GMT +1:00) Madrid,Paris	113	(GMT +10:00) Sydney,Brisbane
54	(GMT +1:00) Amsterdam,Berlin	114	(GMT +10:00) Hobart
55	(GMT +1:00) Bern,Rome	115	(GMT +10:00) Vladivostok
56	(GMT +1:00) Stockholm,Vienna	116	(GMT +10:00) Guam,Port Moresby
57	(GMT +1:00) West Central Africa	117	(GMT +11:00) Solomon Islands
58	(GMT +1:00) Windhoek	118	(GMT +11:00) New Caledonia
59	(GMT +2:00) Bucharest,Cairo	119	(GMT +11:00) Chokurdakh (RTZ 10)
60	(GMT +2:00) Amman,Beirut	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

Time and Date Display

A clock and calendar display on the phones by default. You can choose how to display the time and date for your time zone in several formats, or you can disable the display of the time and date. 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.

To have the most accurate time, 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 the phones in PSTN mode and are set by an incoming call with a supported caller ID standard, or when the phone is connected to Ethernet and you enable the date and time display.

Configuring the Time and Date Display

Use the parameters in the following table to configure time and display options.

Time and Date Display Parameters

Parameter Function	**template** > parameter
Turn the time and date display on or off.	**features.cfg** > up.localClockEnabled
Set the time and date display format.	**site.cfg** > lcl.datetime.date.*
Display time in the 24-hour format.	**site.cfg** > lcl.datetime.time.24HourClock
Set the basic SNTP settings and daylight savings parameters.	**site.cfg** > tcpIpApp.sntp.*

.* indicates grouped parameters. See the section Configuring Phone Groups with the Master Configuration File for more information.

Date Formats

Use the following table to choose values for the `lcl.datetime.date.format` and `lcl.datetime.date.longformat` parameters. The table shows values for Friday, August 19, 2011 as an example.

Date Formats

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

Date Formats

YY/MM/DD	n/a	11/08/19
YYYY/MM/DD	n/a	2011/08/11

Phone Languages

All phones support 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.

Each language is stored as a language file in the **VVXLocalization** folder, which is included with the Polycom UC Software package. 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/>).

At this time, the updater is available in English only.

Setting the Phone Language

You can select the language that displays on the phone using the parameters in the following table.

Phone Language Parameters

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 Configuring Phone Groups with the Master Configuration File for more information.

Add a Language for the Phone Display and Menu

Use the multilingual parameters to add a new language to your provisioning server directory to display on the phone screen and menu.

To add a new language:

1 Create a new dictionary file based on an existing one.

2 Change the strings making sure to encode the XML file in UTF-8 but also ensuring the UTF-8 characters chosen are within the Unicode character ranges indicated in the tables below.

3 Place the file in an appropriately named folder according to the format `language_region` parallel to the other dictionary files under the VVXLocalization folder on the provisioning server.

4 Add an `lcl.ml.lang.clock.menu.x` parameter to the configuration file.

5 Add `lcl.ml.lang.clock.x.24HourClock, lcl.ml.lang.clock.x.format, lcl.ml.lang.clock.x.longFormat`, and `lcl.ml.lang.clock.x.dateTop` parameters and set them according to the regional preferences.

6 (Optional) Set `lcl.ml.lang` to be the new `language_region` string.

Unique Line Labels for Registration Lines

You can configure unique line registrations to display for line keys configured to display on multiple line keys on the phone. For example, you can set different names to display for the registration *4144* that displays on four line keys.

Configuring Unique Line Labels for Registration Lines

When using this feature with the parameter `reg.x.label.y` where 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

Status Indicators on the RealPresence Trio Solution

The RealPresence Trio 8800 and Visual+ systems use LED lights to indicate the status of the solution. The following tables describe each of the status indicators on the RealPresence Trio 8800 and RealPresence Trio Visual+.

RealPresence Trio 8800 Status Indicators

Status	Description
Off	Device is in idle state or powered off.
Green	In a call with audio unmuted.

RealPresence Trio 8800 Status Indicators

Status	Description
Red	Microphones are muted. Device is in a call or in idle state.
Yellow	Power on LED diagnostic.
Amber/Red/Green/Off Repeating	Recovery in progress.

RealPresence Trio Visual+ Status Indicators

Status	Description
Off	Device is not powered on
Flashing red	Device is booting up or pairing
Flashing green	Device update is in progress
Steady green	Device is powered on and paired with the RealPresence Trio 8800
Amber	Device is in a low power, standby state
Alternating orange/red/green/off flashes	Device is in recovery mode
Flashing red	The pairing button has been pressed
Alternating red and green flashes	Device is in pairing diagnostics mode

Example: Set an LED Pattern for Active Calls

In the following example, during an active call, the line key alternates green and red.

To configure a line key LED pattern to alternate green and red for active calls:

» Configure the pattern as follows:

- ➢ `ind.pattern.active.step.1.color=`"Green"
- ➢ `ind.pattern.active.step.1.state=`"1"
- ➢ `ind.pattern.active.step.1.duration=`”1000”
- ➢ `ind.pattern.active.step.2.color=`"Red"
- ➢ `ind.pattern.active.step.2.state=`"1"
- ➢ `ind.pattern.active.step.2.duration=`”1000”

Example: 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.

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

Parameter / Value	Function
`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 the second step in which the LED indicator is off. After this duration, the pattern repeats.

You can turn off the MWI or change the duration of the pattern steps.

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

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

Example: 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.

Parameter / Value	Function
`ind.pattern.offering.step.1.state`=”1”	Turns on the LED indicator.
`ind.pattern.offering.step.1.duration`=”5000”	Sets the duration of the pattern in step 1.
`ind.pattern.offering.step.1.color`=”Yellow”	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`=”5000”	Sets the duration of the pattern in step 2.
`ind.pattern.offering.step.2.color`=”Yellow”	Value is ignored because step 2 state=0
`ind.pattern.offering.step.3.state`=”1”	Turns on the LED indicator.
`ind.pattern.offering.step.3.duration`=”5000”	Sets the duration of the pattern.
`ind.pattern.offering.step.3.color`=”Red”	Sets the color of the LED indicator for the pattern. After this duration, the pattern repeats.

To change the color of the line key indicator:

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

Phone Number Formatting

By default, phone numbers entered on the system are automatically formatted with dashes between dialed numbers following the North American Numbering Plan (NANP), for example: 12223334444 displays as 1-222-333-4444.

Configuring Number Formatting

Use the parameter in the following table to enable or disable number formatting.

Number Formatting Parameters

Parameter Template	Permitted Values
`up.formatPhoneNumbers` template.cfg	1 (default) - Enable automatic number formatting. 0 - Disable automatic number formatting.

Displaying a Number or Custom Label

On the RealPresence Trio, you can choose to display a number, an extension, or a custom label on the Home Screen below the time and date

Configure the Number or Label from the System

You can configure the display of the number or label on the Home screen from the system menu.

To configure the number or label from the RealPresence Trio 8800 system menu:

» Navigate to **Settings > Advanced > Administration Settings > Home Screen Label**.

Configuring the Number and Label Display

You can configure display of the RealPresence Trio 8800 number or label on the Home screen using centralized provisioning parameters.

RealPresence Trio Number and Label Display Parameters

Parameter Template	Permitted Values
`homeScreen.customLabel`	Specify the label to display on the phone's Home screen when `homeScreen.labelType="Custom"`. The label can be 0 to 255 characters. Null (default)
`homeScreen.labelType`	Specify the type of label to display on the phone's Home screen. PhoneNumber (default) • When the phone is set to use Lync Base Profile, the phone number is derived from the Skype for Business server. • When the phone is set to use the Generic Base Profile, the phone uses the number you specify in `reg.1.address`. Custom - Enter an alphanumeric string between 0 and 255 characters. None - Don't display a label.
`homeScreen.labelLocation`	Specify where the label displays on the screen. StatusBar (default) - The phone displays the custom label in the status bar at the top of the screen. BelowDate - The phone displays the custom label on the Home screen only, just below the time and date.

Network and Security Features

After you set up Polycom devices on your network with the default configuration, users can place and answer calls. Polycom's Open SIP UC Software enables you to make custom configurations to optimize security settings.

Wireless Network Connectivity (Wi-Fi)

The RealPresence Trio 8800 supports various wireless modes, security options, radio controls, and Quality of Service monitoring. To ensure the best performance in your location, set a proper country code with the parameter `device.wifi.country` before enabling Wi-Fi.

Enabling Wi-Fi automatically disables the Ethernet port. You cannot use Wi-Fi and Ethernet simultaneously to connect RealPresence Trio 8800 to your network. When you connect the system to your network over Wi-Fi, only audio-only calls are available. Note that RealPresence Trio 8800 does not support Wi-Fi captive portals or Wireless Display (WiDi).

The RealPresence Trio solution supports the following wireless modes:

- 2.4 GHz / 5 GHz operation
- IEEE 802.11a radio transmission standard
- IEEE 802.11b radio transmission standard
- IEEE 802.11g radio transmission standard
- IEEE 802.11n radio transmission standard

You cannot use RealPresence Trio Visual+ for video calls when you connect RealPresence Trio 8800 to your network using Wi-Fi. The RealPresence Trio 8800 and RealPresence Trio Visual+ do not pair when the RealPresence Trio 8800 is connected to your network using Wi-Fi.

Enable Wi-Fi on the RealPresence Trio 8800

You can wirelessly connect the RealPresence Trio 8800 to your network using Wi-Fi, which is disabled by default. When you enable Wi-Fi, the system reboots.

To enable Wi-Fi from the RealPresence Trio 8800:

1 Go to **Settings > Advanced > Administration Settings > Network Configuration > Network Interfaces > Wi-Fi Menu**, and turn Wi-Fi **On**.

The phone restarts.

2 When the phone completes restart, go to **Settings > Advanced > Administration Settings > Network Configuration > Network Interfaces > Wi-Fi Menu** to view available networks.

3 Select a network you want to connect to and press **Connect**.

Configuring Wi-Fi Network Connectivity

The parameters you configure depend on the security mode of your organization and whether or not you enable DHCP. RealPresence Trio 8800 solution is shipped with a security-restrictive worldwide safe Wi-Fi country code setting.

The RealPresence Trio solution supports the following Wi-Fi security modes:

- WEP
- WPA PSK
- WPA2 PSK
- WPA2 Enterprise

Configure Wi-Fi Network Parameters

Parameter Function	**template** > parameter
Enable the Wi-Fi radio.	device.wifi.enabled
Enter the two-letter code for the country in which you enable the Wi-Fi radio.	device.wifi.country
Enable DHCP for Wi-Fi.	device.wifi.dhcpEnabled
	device.wifi.dhcpBootServer
The IP address of the wireless device if not using DHCP.	device.wifi.ipAddress
The network mask address of the wireless device if not using DHCP.	device.wifi.subnetMask
The IP gateway address of the wireless device if not using DHCP.	device.wifi.ipGateway
The SSID of the wireless network.	device.wifi.ssid
Specify the wireless security mode.	device.wifi.securityMode
The length of the hexadecimal WEP key.	device.wifi.wep.key
The hexadecimal key or ASCII passphrase.	device.wifi.psk.key
The EAP to use for 802.1X authentication.	device.wifi.wpa2Ent.method
The WPA2-Enterprise user name.	device.wifi.wpa2Ent.user
The WPA2-Enterprise password.	device.wifi.wpa2Ent.password
	device.wifi.radio.enable2ghz
	device.wifi.radio.enable5ghz

Real-Time Transport Protocol (RTP) Ports

You can configure RTP ports for your environment in the following ways:

- Filter incoming packets by IP address or port.
- Reject packets arriving from a non-negotiated IP address, an unauthorized source, or non-negotiated port for greater security.
- Enforce symmetric port operation for RTP packets. When the source port is not set to the negotiated remote sink port, arriving packets are rejected.
- Fix the phone’s destination transport port to a specified value regardless of the negotiated port.

  This is 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.
- 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, 3550, and 3551, the next-highest odd-numbered port is used to send and receive RTP.

Configuring RTP Ports

Use the parameters in the following table to configure RTP packets and ports.

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

Network Address Translation

Network Address Translation (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. Note that the source port in transmitted packets is the same as the associated listening port used to receive packets.

Network Address Translation Configuration

You can configure the external IP addresses and ports used by the NAT on the phone’s behalf on a per-phone basis. Use the parameters in the following table to configure NAT.

Network Access Translation Parameters

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

Server Redundancy

Server redundancy is often required in VoIP deployments to ensure continuity of phone service if, for example, the call server is taken offline for maintenance, the server fails, or the connection between the phone and the server fails. Polycom phones support Failover and Fallback server redundancy types. In some cases, you can deploy a combination of the two server redundancy types. Consult your SIP server provider for recommended methods of configuring phones and servers for failover configuration.

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

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*.

Configuring Server Redundancy

Use the parameters in the following table to set up server redundancy for your environment.

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.*

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.

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/>.

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`.

Be careful 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.

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.

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

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.

You can statically configure a set of DNS NAPTR SRV and/or A records into the phone.

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 Static DNS

Phones configured with a DNS server behave as follows:

1. 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.
2. 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.
3. After the configured time interval has elapsed, a resolution attempt of the hostname again results in a query to the DNS.
4. 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.

If a phone is not configured with a DNS server, when the phone attempts to resolve a hostname within the static DNS cache, it always returns the results from the static cache.

Use the following table to configure static DNS settings.

Static DNS Cache Parameters

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.*

Static DNS Cache Parameters

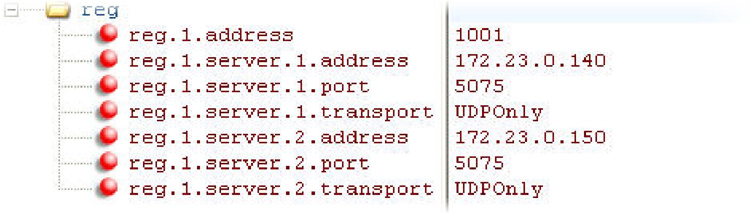
Specify whether to use DNS primary and secondary address set using the parameters `tcpIpApp.dns.server` and `tcpIpApp.dns.altServer`.	**site.cfg** > tcpIpApp.dns.address.overrideDHCP
Specify whether to use the DNS domain name set using the parameter `tcpIpApp.dns.domain`.	**site.cfg** > tcpIpApp.dns.domain.overrideDHCP

.* indicates grouped parameters. See the section Configuring Phone Groups with the Master Configuration File for more information.

Example Static DNS Cache Configuration

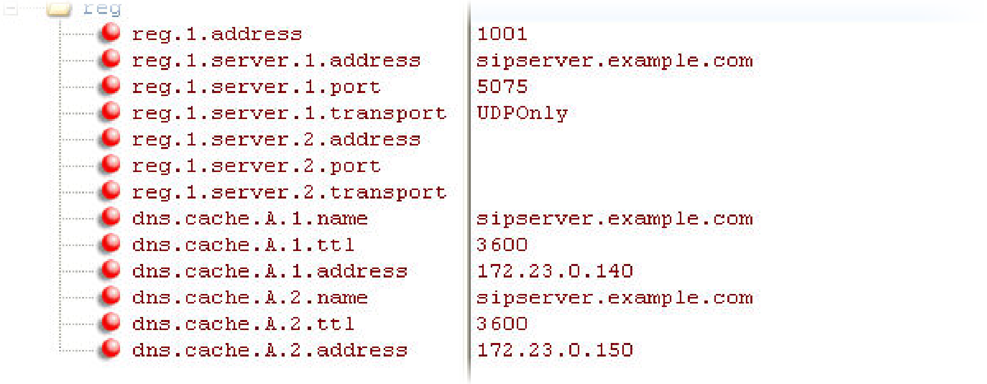
The following example shows how to configure static DNS cache using A records IP addresses in SIP server address fields.

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:



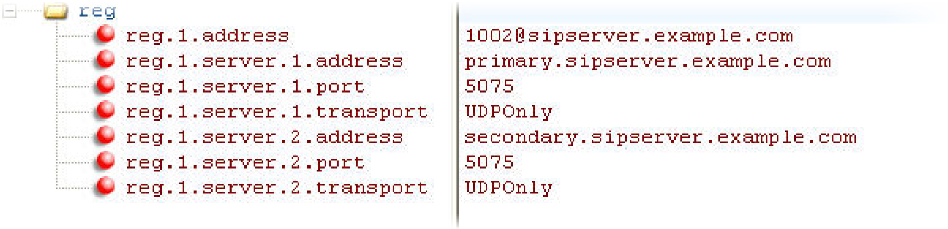
```
reg
  reg.1.address               1001
  reg.1.server.1.address      sipserver.example.com
  reg.1.server.1.port         5075
  reg.1.server.1.transport    UDPOnly
  reg.1.server.2.address
  reg.1.server.2.port
  reg.1.server.2.transport
  dns.cache.A.1.name          sipserver.example.com
  dns.cache.A.1.ttl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

The addresses listed in this example are read by Polycom UC Software in the order listed.

Example 1

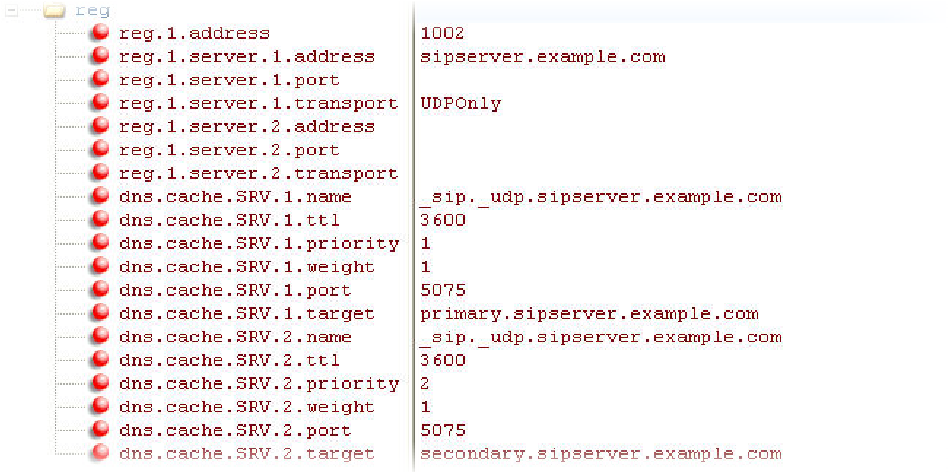
This example shows how to configure static DNS cache where your DNS provides A records for `reg.x.server.x.address` but not SRV. In this case, the static DNS cache on the phone provides SRV records. For more information, see RFC 3263.

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:



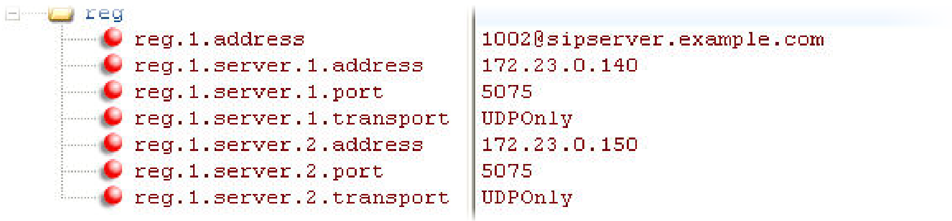
```
reg
    reg.1.address               1002
    reg.1.server.1.address      sipserver.example.com
    reg.1.server.1.port
    reg.1.server.1.transport    UDPOnly
    reg.1.server.2.address
    reg.1.server.2.port
    reg.1.server.2.transport
    dns.cache.SRV.1.name        _sip._udp.sipserver.example.com
    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
```

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

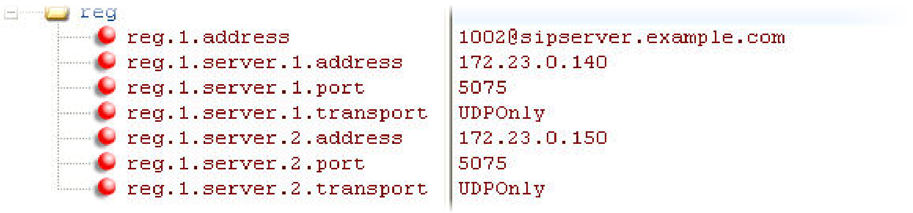
Example 2

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:

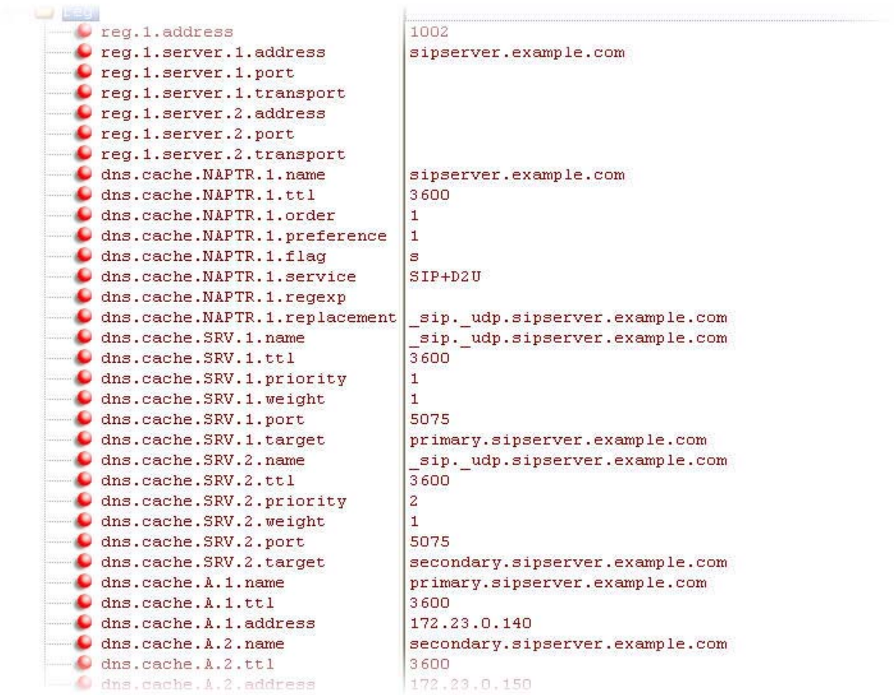


```
reg
    reg.1.address               1002@sipserver.example.com
    reg.1.server.1.address      172.23.0.140
    reg.1.server.1.port         5075
    reg.1.server.1.transport    UDPOnly
    reg.1.server.2.address      172.23.0.150
    reg.1.server.2.port         5075
    reg.1.server.2.transport 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



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

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.

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 regularly change the IP addresses of phones for security purposes. You can configure SIP instance to identify individual phones instead of using IP addresses. This feature complies with RFC 3840.

This feature is not available on VVX 101 and 201 business media phones.

Configuring SIP Instance

The parameter `reg.x.gruu` provides a contact address to a specific user agent (UA) instance, which 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. Refer to the following table for information on configuring this feature.

`Parameter` Template	Permitted Values
`reg.x.gruu` sip-interop.cfg	`1`—The phone sends sip.instance in the REGISTER request. `0` (default)—The phone does not send sip.instance in the REGISTER request.

Provisional Polling of Polycom Phones

You can configure phones to poll the server for provisioning updates automatically, and you can set the phone's automatic provisioning behavior to one of the following:

- **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 set time interval.
  - If the time period is less than or equal to one day, the first poll is at a random time between when the phone starts 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.

Configuring Provisional Polling

Use the parameters in the following table to configure provisional polling.

Note that If `prov.startupCheck.enabled` is set to 0, then Polycom phones do not look for the sip.ld or the configuration files when they reboot, lose power, or restart. 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.

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.*

Example Provisional Polling Configuration

The following are examples of polling configurations you can set up:

- 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.

SIP Subscription Timers

You can configure a subscription expiry independently of the registration expiry. You can also configure an overlap period for a 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.

Configuring SIP Subscription Timers

Use the parameters in the following table to configure when a SIP subscription expires and when expirations overlap.

SIP Subscription Timers

Parameter Function	**template** > parameter
A global parameter that sets the phone’s requested subscription period.	**sip-interop.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.	**sip-interop.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

Administrator and User Passwords

You must have a user or administrator password before you can access certain menu options on the phone and in the Web Configuration Utility. You can use the following default passwords to access menu options on the phone and to access the Web Configuration Utility:

- Administrative password: **456**
- User password: **123**

You can use an administrator password where a user password is required, and you will see the same menu options as the user. If the phone requires the administrator password, you can use the user password, but you are presented with limited menu options.

The Web Configuration Utility displays different features and options depending on which password is used.

Change the Default Administrator Password on the Phone

If you do not change the default administrative password, the phone displays a warning and a reminder message each time the phone reboots. 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.

To change the default administrator password on the phone:

1 On the phone, navigate to **Settings > Advanced**, and enter the default password.

2 Select **Administration Settings > Change Admin Password**.

3 Enter the default password, enter a new password, and confirm the new password.

Change the Default Passwords in the Web Configuration Utility

You can change the administrator and user passwords on a per-phone basis using the Web Configuration Utility. If the default administrative password is in use, a warning displays in the Web Configuration Utility.

To change the default password in the Web Configuration Utility:

1 In the Web Configuration Utility, select **Settings > Change Password**.

2 Update the passwords for the **Admin** and **User**.

Configuring Administrator and User Passwords

Use the parameters in the following table to set the administrator and user password and configure password settings.

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.	**features.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

Disabling External Ports and Features

You can disable unused external phone ports and features to increase the security of devices in your deployment. You can disable the following ports and features:

- **Web Configuration Utility**
- **PC port**
- Aux port
- **USB Port**
- Speakerphone
- Call forwarding
- Do Not Disturb
- Push-to-Talk (PTT)
- **Auto Answer**
- **Applications icon**

At least one audio port must be enabled to send and receive calls.

Configuration for Disabling External Ports and Features

Use the parameters in the following table to disable external ports or specific features.

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.	**features.cfg** > up.handsfreeMode
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
Enable or disable the phone's Autoanswer menu.	**features.cfg** > call.autoAnswerMenu.enable
Enable or disable the Applications icon on the phone's Home screen.	**features.cfg** > homeScreen.application.enable

Visual Security Classification

The security classification of a call is determined by the lowest security classification among all participants connected to a call. For example, a Top Secret classification displays when all participants in a call have a Top Secret classification level.

Call 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. For example, 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.

Phone users can modify their assigned security classification level to a value lower than their assigned level during a call. When the call is over, the server resets the user's classification level to its original state.

Configuring Visual Security Classification

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.

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 a user establishes a call to a phone not connected to this feature, the phone displays as Unclassified.

The following table lists the parameters you can use to configure visual security classification.

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

Incoming Network Signaling Validation

You can choose from the following optional levels of security for validating incoming network signaling:

- Source IP address validation
- Digest authentication
- Source IP address validation and digest authentication

Configuring Network Signaling Validation

The following table includes the parameters you can use to specify the validation type, method, and the events for validating incoming network signaling.

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

Configuration File Encryption

You can encrypt configuration files, contact directories, and configuration override files. You can determine whether encrypted files are the same as unencrypted files and use the SDK to facilitate key generation. You cannot encrypt the master configuration file.

Encrypting Configuration Files

The following table lists the parameters you can use to encrypt configuration 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

Digital Certificates

Polycom phones are installed with a Polycom-authenticated RSA certificate. You can use this certificate to create a secure connection between the phone and server when initiating TLS communications over protocols such as HTTPS and SIP. You can download the Polycom Root CA at http://pki.polycom.com/pki. The certificate is set to expire on March 9, 2044.

X.509 Certificates

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:

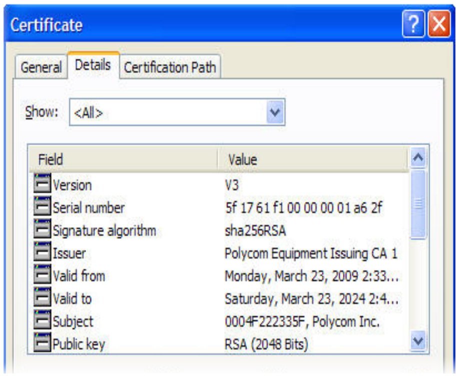
- **Version.** This identifies which version of the X.509 standard applies to this certificate, which in turn affects what information can be specified in the certificate.
- **Serial Number.** The entity that created the certificate is responsible for assigning it a serial number to distinguish it from 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.

Subject Alternative Names

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.



Show: <All>

Field	Value
Version	V3
Serial number	5f 17 61 f1 00 00 00 01 a6 2f
Signature algorithm	sha256RSA
Issuer	Polycom Equipment Issuing CA 1
Valid from	Monday, March 23, 2009 2:33...
Valid to	Saturday, March 23, 2024 2:4...
Subject	0004F222335F, Polycom Inc.
Public key	RSA (2048 Bits)

For more information on digital certificates, see Public Key Infrastructure (X.509) and RFC 2459: Internet X.509 Public Key Infrastructure.

You can install custom device certificates on your Polycom phones in the same way custom CA certificates are installed. See *Technical Bulletin 17877: Using Custom Certificates With Polycom Phones* for more information.

Check for a Device Certificate

The device certificate and associated private key are stored on the phone in its non-volatile memory as part of the manufacturing process. You can check if a phone has a device certificate pre-installed.

To check for a device certificate on a Polycom phone:

1. Navigate to **Settings > Advanced > Administration Settings > TLS Security > Custom Device Credentials**.
2. Select a credential and press the **Info** soft key to view the certificate.

   One of the following messages is displayed:

   - Installed or 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.
   - 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).

- Invalid is displayed if the certificate is not valid.

If your phone reports the device certificate as self-signed rather than Factory installed, return the equipment to receive a replacement.

You can also quickly check if a Polycom device certificate is installed on the phone by navigating to **Settings > Status > Platform > Phone**.

Generating a Certificate Signing Request

If you need a certificate to perform a number of tasks, such as for multiple TLS authentication, you can request a certificate from the phone. By default, the phone requests a 2048-bit certificate with 'sha256WithRSAEncryption' as the signature algorithm. You can use OpenSSL or another certificate signing request utility if you require a stronger certificate.

Obtain a Certificate

The following provides general instructions on how to obtain a certificate.

To obtain a certificate:

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

Generate a Signing Request

The following shows you how to generate a signing request on a Polycom device.

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.
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.

Transport Layer Security 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.

Polycom phones trusts certificates issued by widely recognized certificate authorities when trying to establish a connection to a provisioning server for application provisioning.

Customizing Certificates

You can add custom CA and device certificates to the phone and set up the phone to use the certificates for different features. For example, the phone's factory-installed or custom device certificate can 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.

Determining TLS Platform Profiles or TLS Application Profiles

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 Digital Certificates.

For more information on using custom certificates, see *Technical Bulletin 17877: Using Custom Certificates With Polycom Phones*.

TLS Profile Configuration

By default, all Polycom-installed profiles are associated with the default cipher suite and use trusted and widely recognized CA certificates for authentication.

You can use the parameters in the following table to configure the following TLS Profile feature options:

- Change the cipher suite, CA certificates, and device certificates for the two platform profiles and the six application profiles.
- 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.*

TLS Profile Parameters (continued)

Specify the TLS profile to use for each application (other applications).	**device.cfg** > sec.TLS.profileSelection.*
.* indicates grouped parameters. See the section Configuring Phone Groups with the Master Configuration File for more information.	

TLS Platform Profile and Application Profile Parameters

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	
Specify which CA certificates to use.	**device.cfg** > device.sec.TLS.profile.caCertList1
Specify the cipher suite.	**device.cfg** > device.sec.TLS.profile.cipherSuite1
Select the default cipher suite or a custom cipher suite.	**device.cfg** > device.sec.TLS.profile.cipherSuiteDefault1
Specify a custom certificate.	**device.cfg** > device.sec.TLS.customCaCert1
Specify which device certificates to use.	**device.cfg** > device.sec.TLS.profile.deviceCert1
TLS Application Profile Parameters	
Specify which CA certificates to use.	**site.cfg** > sec.TLS.profile.x.caCert.*
Specify the cipher suite.	**site.cfg** > sec.TLS.profile.x.cipherSuite
Select the default cipher suite or a custom cipher suite.	**site.cfg** > sec.TLS.profile.x.cipherSuiteDefault
Specify a custom certificate.	**site.cfg** > sec.TLS.customCaCert.x
Specify which device certificates to use.	**site.cfg** > sec.TLS.profile.x.deviceCert
Specify the custom device key.	**site.cfg** > sec.TLS.customDeviceKey.x
.* indicates grouped parameters. See the section Configuring Phone Groups with the Master Configuration File for more information.	

Download Certificates to a Polycom Phone

You can download and install up to eight CA certificates and eight device certificates on a Polycom phone. After installing the certificates, you can refresh the certificates when they expire or are revoked, and you can delete any CA certificate or device certificate that you install.

To download a certificate to a Polycom phone:

1. Navigate to **Settings > Advanced > Administrative Settings > TLS Security** and select **Custom CA Certificates** or **Custom Device Certificates**.
2. Select the **Install** soft key.
3. Enter the URL where the certificate is stored.

   For example, *http://bootserver1.polycom.com/ca.crt*

   The certificate is downloaded, and the certificate's MD5 fingerprint displays to verify that the correct certificate is to be installed.
4. Select the **Accept** soft key.

   The certificate is installed successfully.

Mutual Transport Layer Security Authentication

Mutual Transport Layer Security (TLS) authentication is optional and 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 validates the client certificate. If the server is configured to require mutual TLS, a device certificate and an associated private key must be loaded on the phone.

This feature requires that the phone being used has a Polycom factory-installed device certificate or a custom device certificate installed on it. In cases where a phone does 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. For more information, refer to the section Digital Certificates.

The device certificate stored on the phone is used by the following:

- 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)

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.

Polycom Root Certificate Authority

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 lists the tested and verified operating system and web server combinations:

- Microsoft Internet Information Services 6.0 on Microsoft Windows Server 2003

- Apache v1.3 on Microsoft Windows XP

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

You can control which cipher suites to offer and accept during TLS session negotiation. The phone supports the cipher suites listed in the following table. 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

TLS Cipher Suite Configuration

You can use the parameters listed in the following table to configure TLS Cipher Suites.

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) encrypts audio stream(s) to prevent interception and eavesdropping on phone calls. When this feature is enabled, the phones negotiate the type of encryption and authentication to use for the session with the other endpoint.

For more information on SRTP, see RFC 3711. For the procedure describing how two phones set up SRTP for a call, see RFC 4568.

SRTP 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.

If the call is completely secure (RTP authentication and encryption and RTCP authentication and RTCP encryption are enabled), a padlock symbol displays.

Configuring SRTP

Use the session parameters in the following table to turn on or off authentication and encryption for RTP and RTCP streams. You can also turn off the session parameters to reduce the phone's processor usage.

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
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 Configuring Phone Groups with the Master Configuration File for more information.

Phone Lock

This feature enables users to lock their phones to prevent access to menus or directories. If the enhanced feature key (EFK) feature is enabled, you can set a Lock soft key to display on the phone to enable users to quickly lock their phones.

After the phone is locked, users can only place calls to emergency and authorized numbers. You can specify which authorized numbers users can call.

If a user forgets their password, you can unlock the 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.

If a locked phone has a registered shared line, calls to the shared line display on the locked phone and the phone's user can answer the call.

Configuring Phone Lock

Use the parameters in the following table to enable the phone lock feature, set authorized numbers for users to call when a phone is locked, and set scenarios when the phone should be locked.

Phone Lock is different from Device Lock for Skype for Business deployments. If you enable Phone Lock and Device Lock for Skype for Business at the same time on a phone with the Base Profile set to Skype, the Device Lock feature takes precedence over Phone Lock.

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.*
Specify the scenarios when phone lock should be enabled.	**features.cfg** > phoneLock.*

.* indicates grouped parameters. See the section Configuring Phone Groups with the Master Configuration File for more information.

Secondary Port Link Status Report

Polycom devices can detect an externally connected host connection/disconnection, informing the authenticator switch to initiate the authentication process or drop an existing authentication. 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.

This feature ensures the following:

- The port authenticated by the externally attached device switches to unauthenticated upon device disconnection so that other unauthorized devices cannot use it.
- The externally attached device can move to another port in the network and start a new authentication process.
- 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.

- 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.

Configuring the Secondary Port Link Status Report

You can use the parameters in the following table to configure options for the Secondary Port Link Status Report feature, including the required elapse or sleep time between two CDP UPs dispatching.

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

802.1X Authentication

Polycom phones support standard IEEE 802.1X authentication and 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-MD5

Configuring Support for 802.1X Authentication

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 to configure 802.1X Authentication. For more information see Transport Layer Security Profiles.

For more information on EAP authentication protocol, 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

User Profiles

When you set up user profiles, you enable users to access their personal phone settings, including their contact directory, speed dials, and other phone settings from any phone on the network. This feature is particularly useful for remote and mobile workers who do not have a dedicated work space and conduct their business in more than one location. This 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**. If the user profile feature is set up on your company's phones, users can:

- Log in to a phone to access their personal phone settings.
- Place a call to an authorized number from a phone that is in the logged out state.
- Change their user password.
- Log out of a phone after they finish using it.

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 another phone. When a user logs out, the user's personal phone settings are no longer displayed.

You can configure all company 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`.

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 can authenticate their phone by entering correct server credentials.

The phone downloads log files app.log and boot.log from the generic profile on the provisioning server regardless of user logins.

Remotely Logging Out Users

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.

If a user is not logged out from a phone and other users are not prevented 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 and generic credentials on the provisioning server when a user is not logged into the phone.

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.

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 in the Home directory of each user with a user-specific configuration file that you store on the provisioning server with a unique name 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.

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

Configuring User Profiles

To set up the this feature, you need to 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, registration, and other user-specific settings that you want to define.

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.

When you set up the user profile feature, you can determine the following conditions:

- Users are required to always log in to use a phone and access their personal settings.
- Users are not required to log in, and users have the option to use the phone as is without access to their personal settings.
- Users are automatically logged out of the phone when the phone restarts or reboots.
- Users remain logged in to the phone when the phone restarts or reboots.

Use the parameters in the following table to enable users to access their personal phone settings from any phone in the organization.

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 clear text. 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 Default Credentials and a Profile for a Phone

You can choose to define default credentials for a phone, which the phone uses to automatically log itself in each time an actual user logs out or the phone restarts or reboots. When the phone logs itself in using the default login credentials, a default phone profile displays, and users retain the option to log in and view their personal settings.

You can create a new phone configuration file for the default profile, then add and set the attributes for the feature. You can also update an existing phone configuration file to include the user login parameters you want to change.

Polycom recommends that you create a single default user password for all users.

To create default credentials and a profile for a phone:

1 Create a **site.cfg** file for the phone and place it on the provisioning server.

  You can base your file on the sample configuration template 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, then add and set values for the user login attributes.

Create a User Configuration File

Create a configuration file for each user that you want to enable to log in to the phone. The name of the file should specify the user's login ID. In the file, specify any user-specific settings that you want to define for the user.

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.

2 Name each file the ID the user will use 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**.

3 In each **<user>.cfg** file, you can add and set values for the user's login password (optional).

4 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).

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.

If a user updates their password or other user-specific settings 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. The following is a list of configuration parameter precedence (from first to last) for a phone that has the user profile feature enabled:

- <user>-phone.cfg
- Web Configuration Utility
- Configuration files listed in the master configuration file (including <user>.cfg)
- Default values

Third-Party Servers

This section provides information on configuring phones and features with third-party servers.

BroadSoft BroadWorks Server

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 BroadSoft features require you to authenticate the phone with the BroadWorks XSP service interface as described in the section Authentication with BroadWorks Xtended Service Platform (XSP) Service Interface.

Authentication with BroadWorks Xtended Service Platform (XSP) Service Interface

You can configure Polycom phones to use advanced features available on the BroadSoft BroadWorks server. The phones support the following advanced BroadSoft features:

- 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.

Configuring Authentication for BroadWorks XSP

The authentication method you 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`

See the following table for additional details on these parameters.

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

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 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 300, 400, 500 and VVX 600 series business media phones 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 use 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.

Configuring BroadSoft UC-One

You can configure the UC-One Call Settings menu and feature options on the phone, in the Web Configuration Utility, and using configuration parameters.

Configure BroadSoft UC-One on the Phone

You can enable the BroadSoft UC-One feature directly from the phone.

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.

Configure BroadSoft UC-One in the Web Configuration Utility

You can enable the BroadSoft UC-One feature and feature options in the Web Configuration Utility.

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.

BroadSoft UC-One Configuration Parameters

The following table lists all parameters available to configure features in the BroadSoft UC-One application.

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

BroadSoft UC-One Application

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

Anonymous Call Rejection

Anonymous Call Rejection enables users to automatically reject incoming calls from anonymous parties who have restricted their caller identification. 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.

Configuring Anonymous Call Rejection

You can enable the Anonymous Call Rejection feature using configuration files or the Web Configuration Utility. Use the parameters in the following table to enable this feature.

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

Configure Anonymous Call Rejection using the Web Configuration Utility

You can configure Anonymous Call Rejection 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**.

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 display of the Simultaneous Ring menu option on the phone, users can turn the feature on or off from the phone and define which numbers should be included in the Simultaneous Ring group.

Configuring Simultaneous Ring Personal

You can enable or disable the Simultaneous Ring feature for users using configuration files or the Web Configuration Utility. Use the parameters in the following table to enable this feature.

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 display of 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.

Configuring Line ID Blocking

You can configure this feature using configuration parameters or the Web Configuration Utility. Use the parameters in the following table to enable this feature.

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.

Configuring BroadWorks Anywhere

You can configure BroadWorks Anywhere using configuration files or the Web Configuration Utility. Use the parameters in the following table to enable this feature.

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

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.

Configuring Remote Office

You can configure Remote Office using configuration files or the Web Configuration Utility. Use the parameters in the following table to enable this feature.

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
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.

Configuring BroadSoft UC-One Credentials

Use the parameters in the following table to enable this feature.

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

Skype for Business and Lync Server with RealPresence Trio Solution

You can deploy the RealPresence Trio 8800 and Visual+ solution with Microsoft® Skype™ for Business Online, Microsoft® Skype™ for Business 2015, Microsoft® Lync® 2013, and Microsoft® Lync® 2010 on-premises.

For a list of available features and instructions on deploying RealPresence Trio solution with Skype for Business and Lync Server, see the latest *Polycom UC Software with Skype for Business– Deployment Guide* at RealPresence Trio on Polycom Support.

Video Features

After you set up Polycom phones on your network with the default configuration, users can place and answer video calls, if supported. This section provides information on making custom configurations to optimize video calling for Polycom phones. Polycom Open SIP video is compatible with RFC 3984 - RTP Payload Format for H.264 video, and RFC 5168 - XML Schema for Media Control.

The RealPresence Trio 8800 system with a paired RealPresence Visual+ connected to a Logitech C930e camera supports transmission and reception of high quality video images.

Setting the Video Layout on RealPresence Trio Solution

When using the RealPresence Trio Visual+ with monitor, you can set how participants and content display during video calls.

The Gallery View layout is supported for video and content during video calls in standard H.264 video meetings or point-to-point calls.

Video Layout Parameters

Parameter Template	Permitted Values
`video.conf.displayLayout.PIP.peopleMode` new.cfg	Choose what the PIP screen displays. selfView (default) - Display your own video. recentTalker - Display video from the current or most recent talker.
`video.conf.displayLayout.gallery.allowContent` new.cfg	1 (default) - Enable Gallery View layout for video and content. Content is scaled to fit into the 720p window of a gallery window. 0 - Disable Galley View layout. Content displays in a full screen window.
`video.conf.galleryView.overlayTimeout` new.cfg	Set the timer for the participant name overlay on the Visual+ monitor when using the Gallery View. 0 (default) - The overlay does not time out. 0 - 60000 ms

Video Transmission and Camera Options

By default, at the start of a video call, the RealPresence Trio solution with the Logitech C930e camera transmits an RTP encapsulated video stream with images captured from the local camera. Users can stop and start video transmission by pressing the video mute button.

You can use the parameters in the following sections to configure video transmission, the video and local camera view, and video camera options.

Configuring Video Transmission

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

Configuring the Video and Camera View

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
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

Configuring Video Camera Options

Use the parameters in the following table to configure the video camera options.

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 with RealPresence Trio

Use the optional RealPresence Trio Visual+ and Logitech C930e USB Webcam to add video to RealPresence Trio 8800 calls. Polycom supports the following video standards and codecs:

- H.264 advanced video coding (AVC) baseline profile and high profile
- H.264 scalable video coding (SVC) (X-H264UC) and Remote Desktop Protocol (RDP) for desktop and application sharing. (Microsoft only)

To support RealPresence Trio solution video interoperability with Cisco, set the following parameters:

- `video.codecPref.H264HP="0"`
- `video.codecPref.H264HP.packetizationMode0="0"`
- `video.codecPref.H264="0"`

The following table lists video codecs supported by the RealPresence Trio 8800 with the RealPresence Trio Visual+.

Supported Video Codecs

Algorithm	MIME Type	Frame Size	Bit Rate (kbps)	Frame Rate (fps)
H.264	H264/90000		6144 kbps	30
XH264UC			6144 kbps	

Supported Video Interoperability with RealPresence Trio

The following table lists Polycom products that support video interoperability with the RealPresence Trio solution.

Supported Polycom Product Interoperability with RealPresence Trio Solution

Polycom Product	Protocol	Software Version
Polycom® RealPresence® Distributed Media Application™ (DMA®) system	SIP	6.3.1.2-212636
Polycom® RealPresence® Collaboration Server (RMX®) solution	SIP	8.6.4.36
Polycom HDX® 9000 series	SIP/ISDN	3.1.10
Polycom HDX 8000 series	SIP/ISDN	3.1.10
Polycom HDX 7000 series	SIP/ISDN	3.1.10
Polycom HDX 6000	SIP/ISDN	3.1.10
Polycom HDX 4000 series	SIP/ISDN	3.1.10
RealPresence® Group Series 300	H.264/SIP/TIP	5.1.1
RealPresence Group Series 500	H.264/SIP/TIP	5.1.1
RealPresence Group Series 700	H.264/SIP/TIP	5.1.1

Toggling Between Audio-only or Audio-Video Calls

When this feature is enabled on the RealPresence Trio 8800 system using RealPresence Trio Visual+ video capabilities, you can toggle calls between audio-only or audio-video.

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 the phone is registered, you can:

- Use `video.callMode.default` to begin calls as audio-video or audio only. By default, calls begin as audio-video. If you set this parameter to audio, users can press a button on the RealPresence Trio to add video. After a video call has ended, the phone returns to audio-only.
- Use `up.homeScreen.audioCall.enabled` to enable a Home screen icon that allows you to make audio-only calls. Far-end users can add video during a call if the far-end device is video capable.

Configuring Audio-only or Audio-Video Calls

The following parameters configure whether the phone starts a call with audio and video.

Parameter Template	Permitted Values
`up.homeScreen.audioCall.enabled` features.cfg	0 (default) - Disable a Home screen icon that allows users to make audio-only calls. 1 - Enable a Home screen icon that allows users to make audio-only calls. Devices that support video calling show an 'Audio Call' button on the Home screen to initiate audio-only calls.
`video.autoStartVideoTx` video.cfg	1 (default) - Automatically begin video to the far side when you start a call. 0 - Video to the far side does not begin. Note that when the phone Base Profile is set to Skype or Lync, the default is 1.
`video.callMode.default` video.cfg	**RealPresence Trio** Allow the user to begin calls as audio-only or with video. video (default) audio - Set the initial call to audio only and video may be added during a call. On RealPresence Trio solution, you can combine this parameter with `video.autoStartVideoTx`.

Content Sharing

You can show content from a computer during in-person meetings, video conference calls, and point-to-point video calls on the RealPresence Trio Visual+ system monitor. To share content:

- The RealPresence Visual+ system must be paired with the RealPresence Trio 8800 system
- The computer and RealPresence Trio solution must be able to communicate on the same IP network

You can use the following Polycom applications to share content:

- Polycom® People+Content® (PPCIP)
- Polycom® RealPresence® Desktop for Windows® or Mac®
- Polycom® RealPresence® Mobile application

You can download People+Content IP and RealPresence Desktop from Polycom Support and RealPresence Mobile from your mobile application store.

For information about using PPCIP on the RealPresence Trio solution registered with Skype for Business, see the *Polycom RealPresence Trio - User Guide* at RealPresence Trio on Polycom Support.

The default port used by Group Paging when enabled conflicts with the UDP port 5001 used by Polycom® People+Content™ on the RealPresence Trio system. Since the port used by People+Content is fixed and cannot be configured, configure one of the following workarounds:

- Configure a different port for Group Paging using parameter `ptt.port`.
- Disable People+Content IP using parameter `content.ppcipServer.enabled="0"`.

Configuring Content Sharing

Use the parameters in the following table to configure content sharing options.

To enable device pairing with the RealPresence Trio solution, use the `smartpairing*` parameters. Note that People+Content IP does not support ultrasonic SmartPairing.

Content Sharing Parameters

Parameter Template	Permitted Values
`content.autoAccept.rdp` new.cfg	1 (default) - Content shown by far-end users is automatically accepted and displayed on the RealPresence Trio solution. 0 - Near-end users are prompted to accept meeting content sent to RealPresence Trio solution from a far-end user.
`content.bfcp.port`	15000 (default) - 0 - 65535 -
`content.bfcp.transport`	UDP (default) - TCP -
`content.ppcipServer.enabled`	1 (default) - Enable Polycom People+Content IP. 0 - Disable Polycom People+Content IP.
`content.ppcipServer.meetingPassword`	NULL (default) - String (0 - 256 characters) -
`smartPairing.mode`	Enables users with People+Content IP or RealPresence Desktop on a computer or RealPresence Mobile on a tablet to pair with the RealPresence Trio 8800 conference phone using SmartPairing. Disabled (default) - Users cannot use SmartPairing to pair with the conference phone. Manual - Users must enter the IP address of the conference phone to pair with it.
`smartPairing.volume`	The relative volume to use for the SmartPairing ultrasonic beacon. 6 (default) 0 - 10

Polycom People+Content IP over USB

You can use Polycom® People+Content® IP (PPCIP) to share video or data from a Windows® or Mac® computer connected by USB to the RealPresence Trio 8800 system when in or out of a call. When you install PPCIP version 1.4.2 and run it unopened in the background, the PPCIP application pops up immediately when you connect the computer to RealPresence Trio solution via USB.

Keep the following points in mind:

- Showing content with People+Content IP over USB provides content to a maximum of 1080p resolution on a connected Windows or Mac computer.
- Audio content is not shared.

- Content sent from People+Content is sent over USB, and no network connection is needed. This is useful for environments where guest IP access is not allowed. You can show content with People+Content IP on a computer connected by USB to RealPresence Trio to a maximum of 1080p resolution on a Windows computer. You must use UC Software 5.4.3AA or later to share your desktop at up to 1080p resolution using a Mac computer connected by USB to the RealPresence Trio solution.

Configuring Polycom People+Content IP over USB

This following table lists parameters that configure the People+Content over USB feature.

Polycom People+Content Content Sharing Parameters

Parameter Template	Permitted Values
`feature.usb.device.content`	1 (default) - Enable content sharing using the People+Content IP application on a computer connected by USB to RealPresence Trio solution. 0 - Disable content sharing using the People+Content IP application on a computer connected by USB to RealPresence Trio solution.

Polycom People+Content IP

You can share content from a computer over IP using Polycom® RealPresence® Desktop Software, Polycom® People+Content IP (PPCIP), and Polycom® RealPresence® Mobile Software. Sharing content with Polycom People+Content IP from a computer connected over IP supports 1080p resolution with about five frames per second on the RealPresence Visual+ monitor. The computer and RealPresence Trio solution must be able to communicate on the same IP network and you must pair your Polycom software application with the RealPresence Trio system.

When RealPresence Trio is registered with Skype for Business, you can use these applications to share content only to a local monitor. You cannot share content from a RealPresence Trio system over a Skype for Business call. For instructions, see the *Polycom RealPresence Trio - User Guide* at RealPresence Trio on Polycom Support.

Configuring RealPresence Trio Solution Content Sharing

The following table lists parameters that configure content sharing with the RealPresence Trio solution.

Polycom People+Content IP Parameters

Parameter Template	Permitted Values
`content.autoAccept.rdp` new.cfg	1 (default) - Content shown by far-end users is automatically accepted and displayed on the RealPresence Trio solution. 0 - Near-end users are prompted to accept meeting content sent to RealPresence Trio solution from a far-end user.
`content.bfcp.port`	15000 (default) 0 - 65535

Polycom People+Content IP Parameters

Parameter Template	Permitted Values
`content.bfcp.transport`	UDP (default TCP
`content.ppcipServer.enabled`	1 (default) - Enable Polycom People+Content IP. 0 - Disable Polycom People+Content IP.
`content.ppcipServer.meetingPassword`	NULL (default) String (0 - 256 characters)

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.forceRtcpVideoCodecControl	video.dynamicControlMethod	voIpProt.SDP.offer.rtcpVideoCodecControl	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.

I-Frame Parameter Dependencies (continued)

video. forceRtcpVideoCodecControl	video. dynamicControlMethod	voIpProt. SDP.offer.rtcpVideoCodecControl	Behavior when requesting video I-frame updates
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.
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.

<acd/>

The following table shows <acd/> 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.

<apps/>

The following table lists <apps/> 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.

<attendant/>

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. 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	**0 or 1**	**1**
Normal		
attendant.behaviors.display.spontaneousCallAppearances.automata	**0 or 1**	**0**
Automatic		
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	**0 or 1**	**1**
Normal		
attendant.behaviors.display.remoteCallerID.automata		
Automatic		

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.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.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.

<bluetooth/>

The following table specifies the Bluetooth parameter for the RealPresence Trio 8800 and VVX 600/601 phone.

Bluetooth Radio Transmitter Parameter

Parameter	Permitted Values	Default
bluetooth.devName	**UTF-8 string**	**NULL**
Enter the name of the device that broadcasts over Bluetooth to other devices.		
bluetooth.discoverableTimeout	**0 - 3600**	**0**
Set the time in seconds after which other devices can discover this device over Bluetooth. If set to 0, other devices can always discover this device over Bluetooth.		
bluetooth.pairedDeviceMemorySize	**0 – 10**	**10**
bluetooth.radioOn	**0 or 1**	**0**
If 0, the Bluetooth radio is off. If 1, the Bluetooth radio is on. The Bluetooth radio must be turned on before other devices can connect to this device over bluetooth		

<call/>

The phone supports an optional per-registration feature that enables automatic call placement when the phone is off-hook.

The phone supports a per-registration configuration that determines which events 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.

Call Parameters

Parameter	*Permitted Values*	*Default*
call.advancedMissedCalls.addToReceivedList	**0 or 1**	**0**
Applies to calls on that are answered remotely. If 0, calls answered from the remote phone are not added to the local receive call list. If 1, calls answered from the remote phone are added to the local receive call list.		
call.advancedMissedCalls.enabled	**0 or 1**	**1**
If 1, improved missed call handling for shared lines is enabled (shared lines can correctly count missed calls). If 0, the old missed call handling is used for shared lines (shared lines may not correctly count missed calls).		
call.advancedMissedCalls.reasonCodes	**comma-separated list of indexes**	**200**
A comma separated list of reason code indexes that are interpreted to mean that a call should not be considered as a missed call.		
call.autoAnswer.micMute	**0 or 1**	**1**

Call Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.autoOffHook.x.enabled[1]	**0 or 1**	**0**
Enable or disable the feature		
call.autoOffHook.x.contact[1]	**a SIP URL**	**Null**
The contact address to where the call is placed		
call.autoOffHook.x.protocol[1]	**SIP**	**Null**
The calling protocol to use		
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.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. 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 Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.defaultTransferType	**Consultative or Blind**	**Generic SIP = Consultative** **Skype = 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.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 Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.internationalPrefix.key	**0 or 1**	**0**
template: site.cfg		
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.playLocalRingBackBeforeEarlyMediaArrival	**0 or 1**	**1**
template: sip-interop.cfg		
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 Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.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.		
call.stickyAutoLineSeize[1]	**0 or 1**	**0**

Call Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.		
call.stickyAutoLineSeize.onHookDialing[1]	**0 or 1**	**0**
If `call.stickyAutoLineSeize` is set to 1, this parameter has no effect. The regular stickyAutoLineSeize behavior is followed. If `call.stickyAutoLineSeize` is set to 0 and this parameter is set to 1, this overrides the stickyAutoLineSeize behavior for hot dial only. (Any new call scenario seizes the next available line.) If `call.stickyAutoLineSeize` is set to 0 and this parameter is set to 0, there is no difference between hot dial and new call scenarios. Note: A hot dial occurs on the line which is currently in the call appearance. Any new call scenario seizes the next available line.		
call.teluri.showPrompt template: site.cfg	**0 or 1**	**1**
call.switchToLocalRingbackWithoutRTP template: sip-interop	**0 or 1**	**0**
call.urlModeDialing[1]	**0 or 1**	**0**
If 0, URL dialing is disabled. If 1, URL dialing is enabled.		

[1] Change causes phone to restart or reboot.

<callLists/>

The call lists (or call log) parameters 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**

Call List (Call Log) Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.		

<device/>

The `<device/>` parameters—also known as device settings—contain default values that you can use to configure basic settings for multiple phones.

Web Info: Default device parameter values

The default values for the `<device/>` parameters are set at the factory when the phones are shipped. For a list of the default values, see the latest 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.xxx[1]	**string**	
Configuration parameter.		
device.xxx.set[1]	**0 or 1**	**0**

Device Parameter Types (continued)

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		**NULL**
NULL (default) Generic - Disables the Skype for Business graphic interface. Lync - Use this Base Profile for Skype for Business deployments. SkypeUSB - Use this Base Profile when you want to connect RealPresence Trio to a Microsoft Room System or a Microsoft Surface Hub.		
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**	

Device Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.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 Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.net.dot1x.method	**EAP-None, EAP-TLS, EAP-PEAPv0-MSCHAPv2, 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.lldpFastStartCount	**3 to 10**	**5**
Specify the number of consecutive LLDP packets the phone sends at the time of LLDP discovery. Note that LLDP packets are sent every one second.		
device.net.IPgateway[1]	**IP address**	
The phone's default router.		

Device Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.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 Parameters (continued)

Parameter	*Permitted Values*	*Default*
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**	
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 Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.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 Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.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.		
device.wifi.dhcpBootServer	**0, 1, 2, V4, V6, Static**	**0**
device.wifi.dhcpEnabled	**0 or 1**	**0**
If 0, DHCP is disabled for Wi-Fi. If 1, DHCP is enabled for Wi-Fi.		

Device Parameters (continued)

Parameter	*Permitted Values*	*Default*
device.wifi.enabled If 1, the Wi-Fi radio is enabled. If 0, the Wi-Fi radio is disabled.	**0 or 1**	**0**
device.wifi.ipAddress The IP address of the wireless device if you are not using DHCP.	**String**	**0.0.0.0**
device.wifi.ipGateway The IP gateway address for the wireless interface if not using DHCP.	**String**	**0.0.0.0**
device.wifi.psk.key The hexadecimal key or ASCII passphrase.	**String**	**0xFF**
device.wifi.radio.band2_4GHz.enable Enable or disable the 2.4 GHz band for Wi-Fi.	**0 or 1**	**0**
device.wifi.radio.band5GHz.enable Enable or disable the 5 GHz band for Wi-Fi.	**0 or 1**	**0**
device.wifi.securityMode Specify the wireless security mode.	**None, WEP, WPA-PSK, WPA2-PSK, WPA2-Enterprise**	**NULL**
device.wifi.ssid The Service Set Identifier (SSID) of the wireless network.	**SSID**	**SSID1**
device.wifi.subnetMask The network mask address of the wireless device if not using DHCP.	**String**	**255.0.0.0**
device.wifi.wep.key1 The length of the hexadecimal WEP key.	**0 = 40-bits** **1 = 104-bits**	**0**
device.wifi.wpa2Ent.caCert.name Specify the CA certificate alias for Wi-Fi enterprise (EAP) level security. To use the default certificate, set the value to Polycom 802.1X Device Certificate.	**String (0 - 128 characters)**	**NULL**
device.wifi.wpa2Ent.clientCert.name	**String (0 - 128 characters)**	**NULL**

Device Parameters (continued)

Parameter	*Permitted Values*	*Default*
Specify the user or device certificate alias for Wi-Fi enterprise (EAP) level security. To use the default certificate, set the value to Polycom 802.1X Device Certificate. DG		
device.wifi.wpa2Ent.method	**EAP-PEAPv0/MSCHAPv2, EAP-FAST** **EAP-TLS** **EAP-PEAPv0-GTC** **EAP-TTLS-MSCHAPv2** **EAP-TTLS-GTC** **EAP-PEAPv0-NONE** **EAP-TTLS-NONE** **EAP-PWD**	**NULL**
The Extensible Authentication Protocol (EAP) to use for 802.1X authentication.		
device.wifi.wpa2Ent.password		
The WPA2-Enterprise password.		
device.wifi.wpa2Ent.user	**String**	
The WPA2-Enterprise user name.		

[1] Change causes phone to restart or reboot.

<diags/>

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.		
diags.pcap.remote.password	**alphanumeric**	**<MAC Address>**
Enter the remote packet capture password.		

Remote Packet Capture Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.

<dialplan/>

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.		
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**

Dial Plan (Digit Map) Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.		
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.		
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.		
dialplan.impossibleMatchHandling[1]	**0, 1 or 2**	**0**
This parameter applies to digits you enter in dial mode which becomes active after you press the line 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.		
dialplan.removeEndOfDial[1]	**0 or 1**	**1**
If set to 1, strip trailing # digit from digits sent out.		
dialplan.routing.emergency.outboundIdentity	**10-25 digits, 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"		
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.		

Dial Plan (Digit Map) Parameters (continued)

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.		
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.		
dialplan.routing.server.x.port[1]	**1 to 65535**	**5060**
The port of a SIP server to use for routing calls.		
dialplan.routing.server.x.transport[1]	**DNSnaptr, TCPpreferred, UDPOnly, TLS, TCPOnly**	**DNSnaptr**
The DNS lookup of the first server to be dialed 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.		
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.		
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.

Per-Registration Dial Plan (Digit Map) Parameters

Parameter	Permitted Values	Default
dialplan.conflictMatchHandling	0 or 1	Generic Profile=0 Skype 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
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

Per-Registration Dial Plan (Digit Map) Parameters (continued)

Parameter	Permitted Values	Default
dialplan.userDial.timeOut	0 – 99 seconds	Generic Profile=0 Skype Profile=3

[1]Change causes phone to restart or reboot.

<dir/>

This parameter definition includes:

- <broadsoft/> Polycom BroadSoft UC-One directory definitions
- <corp/> The corporate directory definition
- <local/> The local directory definition

<broadsoft/>

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`.		
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.		

<corp/>

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.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.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, other**	**last_name**
Defines how parameter x is interpreted by the phone. Entries can have multiple parameters of the same type. The value other is used for display purposes only. If the user saves the entry to the local contact directory on the phone, `first_name, last_name`, and `phone_number` are copied. The user can place a call to the `phone_number` and `SIP_address` from the corporate directory.		
dir.corp.autoQuerySubmitTimeout[1]	**0 to 60 seconds**	**0**
The timeout (in seconds) between when the user stops entering characters in the quick search and when the search query is automatically submitted. If 0, there is no timeout (automatic submit is disabled).		
dir.corp.backGroundSync[1]	**0 or 1**	**0**
If 0, background downloading from the LDAP server is disabled. If 1, background downloading is enabled.		

Corporate Directory Parameters (continued)

Parameter	*Permitted Values*	*Default*
dir.corp.backGroundSync.period[1]	**3600 to 604800**	**86400**
The corporate directory cache is refreshed after the corporate directory feature has not been used for this period of time seconds. The default period is 24 hours (86400 seconds). The minimum is 1 hour and the maximum is 7 days.		
dir.corp.baseDN[1]	**UTF-8 encoded string**	**Null**
The base domain name. This is the starting point for making queries on the LDAP server.		
dir.corp.bindOnInit[1]	**0 or 1**	**1**
If 0, do not use bind authentication on initialization. If 1, use bind authentication on initialization.		
dir.corp.cacheSize[1]	**32 to 256** **VVX 101=32-64**	**128** **VVX 101=64**
The maximum number of entries that can be cached locally on the phone. Note that the default value and permitted values differ when `dir.corp.cacheSize` is used with the VVX 101 business media 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** **VVX 101=8-32**	**32** **VVX 101=16**
The maximum number of entries requested from the corporate directory server with each query. Note that the default value and permitted values differ when `dir.corp.pageSize` is used with the VVX 101 business media phone.		
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.

<local/>

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]	**RealPresence Trio = 2000-3000** **VVX 300/301/310/311, 400/401/410/411, 500/501, 600/601 = 1-500** **VVX1500 = 1-9999** **SoundStructure VoIP Interface = not applicable**	**RealPresence Trio=2000** **VVX=500** **VVX 1500=9999**
Set the maximum number of contacts allowed in the local contact directory. Note that configuring more than about 1,000 contacts results in slow phone performance for the first minute or two after reboot.		
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 an 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.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.		

[1] Change causes phone to restart or reboot.

<divert/>

The phone has a flexible call forward/diversion feature for each registration. In all cases, a call is diverted only if a non-Null contact has been configured.

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] **divert.busy.x.contact[1]**	**0 or 1** **contact address**	**1** **Null**
Divert incoming calls that reach a busy signal. If `enabled` is set to 1, calls 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] **divert.dnd.x.contact[1]**	**0 or 1** **contact address**	**0** **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] **divert.noanswer.x.contact[1]** **divert.noanswer.x.timeout[1]**	**0 or 1** **contact address** **positive integer**	**1** **Null** **55**
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.

<dns/>

The <dns/> parameters include:

- DNS-A
- DNS-NAPTR
- DNS-SRV

You can enter a maximum of 12 record entries for DNS-A, DNS-NAPTR, and DNS-SRV. records.

DNS-A

Add up to 12 DNS-A record entries using the parameters in the 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**

DNS-NAPTR Parameters (continued)

Parameter	*Permitted values*	*Default*
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.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**

DNS-SRV Parameters (continued)

Parameter	*Permitted values*	*Default*
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.		

<efk/>

Use the following tables to configure the enhanced feature key (EFK) feature on your phone:

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.		

Enhanced Feature Key (EFK) List Parameters (continued)

Parameter Name	*Permitted Values*	*Default*
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.		
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.

<feature/>

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 Activation/Deactivation Parameters (continued)

Parameter	*Permitted Values*	*Default*
feature.acdLoginLogout.enabled[1]	**0 or 1**	**0**
If 0, the ACD login/logout feature is disabled. If 1, the feature is enabled.		
feature.acdPremiumUnavailability.enabled[1]	**0 or 1**	**0**
If 0, the premium ACD unavailability feature is disabled. If 1, premium ACD unavailability feature is enabled, and unavailability reason codes can be used (if the other ACD feature parameters are also be enabled).		
feature.acdServiceControlUri.enabled[1]	**0 or 1**	**0**
If 0, the ACD service control URI feature is disabled. If 1, the feature is enabled.		
feature.bluetooth.enabled	**0 or 1**	**1**
RealPresence Trio 8800 and VVX 600/601 high-security environments. If 0, the Bluetooth feature is disabled. If 1, Bluetooth is enabled. When enabled, the Bluetooth menu shows in the RealPresence Trio 8800 user interface.		
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.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 Activation/Deactivation Parameters (continued)

Parameter	*Permitted Values*	*Default*
feature.CallCenterCallInformation.enable	**0 or 1**	**1**
1 (default) – The phone displays call center and incoming call information in a pop-up message. 0 - The phone does not display call center and incoming call information in a pop-up message.		
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]	**0 or 1**	**1**
All locally controlled call lists.		
feature.callListMissed.enabled[1]	**0 or 1**	**1**
The missed calls list.		
feature.callListPlaced.enabled[1]	**0 or 1**	**1**
The placed calls list.		
feature.callListReceived.enabled[1]	**0 or 1**	**1**
The received calls list.		
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.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.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**

Feature Activation/Deactivation Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.flexibleLineKey.enable	**0 or 1**	**0**
Enables and disables the 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.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.intercom.enable	**0 or 1**	**0**
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.nfc.enabled	**0 or 1**	**1**
If 1, NFC pairing is enabled and users can pair NFC-capable devices to the RealPresence Trio 8800 solution. If 0, NFC pairing is disabled.		
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**

Feature Activation/Deactivation Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.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.usb.device.enabled	**0 or 1**	**1**
The USB device port enables you to use RealPresence Trio 8800 as an audio device for your laptop. If 1, the USB device port is enabled. If 0, the USB device port is disabled. When you disable the RealPresence Trio system's USB device port using the parameter `feature.usb.device.enabled`, the USB Connections settings do not display on the phone menu at Settings > Advanced > Administration Settings > USB Computer Connections.		
feature.usb.host.enabled	**0 or 1**	**1**
If 1, the USB host port is enabled. If 0, the USB host port is disabled. Use the host port for memory sticks, mouse, keyboards, and charging your devices if enabled.		
feature.usbRear.power.enabled	**0 or 1**	**1**

Feature Activation/Deactivation Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.VVXD60.allowLineMappings	**0 or 1**	**0**
0 (default) - The ‘Map Lines’ menu is available only as a password-protected option in the Administrator menu and administrators can map lines on VVX phones to the Polycom D60 handset. 1 - The ‘Map Lines’ menu is available to administrators and to users on VVX phones at Menu > Settings > Features > VVX D60 Configuration to map lines on VVX phones to the Polycom D60 handset.		
device.wifi.enabled	**0 or 1**	**0**
If 1, Wi-Fi is used instead of wired Ethernet for VoIP calls. If 0, Wi-Fi is disabled. Note that you cannot use RealPresence Trio Visual+ for video when using Wi-Fi.		

[1] Change causes phone to restart or reboot.

<freerdp/>

Parameter	*Permitted Values*	*Default*
freerdp.ui.hideTouchPointer	**0 or 1**	**1**

<homescreen/>

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.		

<httpd/>

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** **Skype=0**
Base Profile = Generic 1 (default) - The web server is enabled. 0 - The web server is disabled. Base Profile = Skype 0 (default) - The web server is disabled. 1 - The web server is enabled.		
httpd.cfg.enabled[1]	**0 or 1**	**Generic=1** **Skype=0**
Base Profile = Generic 1 (default) - The Web Configuration Utility is enabled. 0 - The Web Configuration Utility is disabled. Base Profile = Skype 0 (default) - The Web Configuration Utility is disabled. 1 - The Web Configuration Utility is enabled.		

HTTPD (Web Server) Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.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**
1 (default) - Access to the Web Configuration Utility is allowed only over a secure tunnel (HTTPS) and non-secure (HTTP) is not allowed. 0 - Access to the Web Configuration Utility is allowed over both a secure tunnel (HTTPS) and non-secure (HTTP).		

[1] Change causes phone to restart or reboot.

<homeScreen/>

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**
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.newCall.enable	**0 or 1**	**1**
Enable or disable display of the New Call icon on the phone Home screen.		

Homescreen Parameters (continued)

homeScreen.redial.enable	**0 or 1**	**1**
Enable or disable display of the Redial menu 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.		

<lcl/>

You can configure the language you want the Polycom phone user interface to operate and display in. The phones support both North American and international time and date formats.

Caution: Use a multilingual XML editor

Edit the language parameters using a multilingual XML editor. If you do not use an XML editor, some of the language labels in the configuration file and in the language menu on the phone 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:

- Multilingual definitions
- Date and time definitions

<ml/>

The multilingual parameters listed in the following table are based on string dictionary files downloaded from the provisioning server. These files are encoded in XML format and 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**

Multilingual Parameters (continued)

Parameter	*Permitted Values*
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.enabled[1]	**0 or 1**
This parameter applies to RealPresence Trio, 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** **lcl.ml.lang.menu.x.label**[1] **Phone language menu label**	**String in the format language_region** **String in the format nativeLanguageName (abbreviation)**
The phone supports multiple languages. Dictionary files and labels must be sequential (for example, `lcl.ml.lang.menu.1, lcl.ml.lang.menu.2, lcl.ml.lang.menu.3… lcl.ml.lang.menu.N)` The dictionary file cannot have caps, and the strings must exactly match a folder name of a dictionary file (you can find the names in the **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.

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

<datetime/>

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.		
lcl.datetime.date.format	**string which includes ‘D’, ‘d’ and ‘M’ and two optional commas**	
Controls format of date string. D = day of week, d = day, M = month. Up to two commas may be included. For example: D,dM = Thursday, 3 July or Md,D = July 3, Thursday The field may contain 0, 1 or 2 commas which can occur only between characters and only one at a time. For example: “D,,dM” is illegal.		
lcl.datetime.date.longFormat	**0 or 1**	
If set to 1, display the day and month in long format (Friday/November), otherwise, use abbreviations (Fri/Nov).		
lcl.datetime.time.24HourClock	**0 or 1**	
If set to 1, display time in 24-hour clock mode rather than a.m./p.m.		

<license/>

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.

<lineKey/>

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.		
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.

<log/>

The event logging system supports the classes of events listed in the table Logging Levels. Two types of logging are supported:

- level, change, and render
- <sched/>

Caution: Changing the logging parameters
Logging parameter changes can impair system operation. Do not change any logging parameters without prior consultation with Polycom Technical Support.

Logging Levels

Logging Level	*Interpretation*
0	Debug only
1	High detail class event
2	Moderate detail event class
3	Low detail event class

Logging Levels (continued)

4	Minor error—graceful recovery
5	Major error—will eventually incapacitate the system
6	Fatal error

Each event in the log contains the following fields separated by the | character:

- time or time/date stamp
- 1-5 character component identifier (such as “so”)
- event class
- cumulative log events missed due to excessive CPU load
- free form text - the event description

Three formats available for the event timestamp are listed in the 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, 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.apps	**0 - 6**	**4**
Initial logging level for the Apps log module.		
log.level.change.bfcp	**0 - 6**	**4**
Initial logging level for the BFCP content log module.		
log.level.change.dect	**0-6**	**4**
Sets the logging detail level for the VVX D60 accessory.		

Logging Level, Change, and Render Parameters (continued)

Parameter	*Permitted Values*	*Default*
log.level.change.fec	0 - 6	4
Set the log level for video FEC.		
log.level.change.fecde	**0 - 6**	**4**
log.level.change.fecen	**0 - 6**	**4**
log.level.change.flk	**0 - 6**	**4**
Sets the log level for FLK logs.		
log.level.change.mr	**0 - 6**	**4**
Initial logging level for the Networked Devices log module.		
log.level.change.mrcam	**0 - 6**	**4**
Initial logging level for the Networked Devices Camera log module.		
log.level.change.mrcon	**0 - 6**	**4**
Initial logging level for the Networked Devices Connection log module.		
log.level.change.mraud	**0 - 6**	**4**
Initial logging level for the Networked Devices Audio log module.		
log.level.change.mrdis	**0 - 6**	**4**
Initial logging level for the Networked Devices Display log module.		
log.level.change.mrmgr	**0 - 6**	**4**
Initial logging level for the Networked Devices Manager log module.		
log.level.change.pec	**0 - 6**	**4**
Initial logging level for the Polycom Experience Cloud (PEC) log module.		
log.level.change.ppcip	**0 - 6**	**4**
Initial logging level for the People+Content IP 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.render.file	**0 or 1**	**1**

Logging Level, Change, and Render Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.		
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.		

<sched/>

The phone can be configured to schedule certain advanced logging tasks on a periodic basis. Polycom recommends that you set the parameters listed in the 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		
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.		

<mr/>

The next table lists parameters that configure the modules with which the RealPresence Trio 8800 system is to pair. Normally, the UID is the Ethernet MAC address of the RealPresence Trio Visual+ system.

Module Pairing Parameters

Parameter	*Permitted Values*	*Default*
mr.audio.srtp.require	**0 or 1**	**1**
When enabled, SRTP is used to encrypt and authenticate modular room audio signals sent between RealPresence Trio 8800 and RealPresence Trio Visual+..		
mr.bg.selection	HallstatterSeeLake, Auto, BlueGradient, BavarianAlps, ForgetMeNotPond, Custom	HallstatterSeeLake

Module Pairing Parameters (continued)

Parameter	*Permitted Values*	*Default*
Set the background image for the paired RealPresence Trio Visual+ display. HallstatterSeeLake (default) Auto - Automatically cycles through background images 2, 3, 4. The background image changes each time a video call ends. BlueGradient BavarianAlps ForgetMeNotPond Custom - Use a custom background specified by `mr.bg.url`.		
mr.bg.showPlcmLogo	**0 or 1**	**1**
Indicates whether the Polycom logo should be shown on the TV attached to the paired RealPresence Trio 8800 Visual+. By default, the Polycom logo is shown, but it may be hidden by configuring RealPresence Trio with mr.bg.showPlcmLogo="0".		
mr.bg.url	**String (maximum 256 characters)**	
Specifies an HTTP URL of a background image to use on the TV attached to the paired RealPresence Trio 8800 Visual+. This background image will be used only if mr.bg.selection="5".		
mr.pairButton.notification	**0 or 1**	**0**
If 0, the RealPresence Trio 8800 is not notified when you press the Pair button on the RealPresence Trio Visual+. If 1, the RealPresence Trio 8800 is notified when you press the Pair button on the RealPresence Trio Visual+, and the pairing icon displays on status bar of the RealPresence Trio 8800 LCD.		
mr.pair.tls.enabled	**0 or 1**	**1**
If 1, use TLS for communications between the RealPresence Trio 8800 and RealPresence Trio Visual+ systems. If 0, do not use TLS for communications.		
mr.pair.uid.1	**String**	**Null**
Enter the MAC address (Serial Number [S/N]) of the RealPresence Trio Visual+ with which you want to pair.		
mr.video.camera.focus.auto	**0 or 1**	**0**
If 1, the camera's automatic focus is enabled. If 0, the camera's automatic focus is disabled. Automatic focus is not recommended for group call settings.		
mr.video.camera.focus.range	**0 - 255 millimeters**	**0**
Specifies the distance to the camera's optimally-focused target.		
mr.video.iFrame.minPeriod	**1 – 60**	**2**
Minimum period allowed between transmitted video i-Frames or transmitted i-Frame requests.		

<msg/>

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.

<mwi/>

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*

Message Waiting Indicator Parameters (continued)

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.		

<nat/>

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.

<net/>

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.interface.mtu6	**1280 - 1500**	**1500**
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.		

<phoneLock/>

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	**String**	
The name or description of an authorized number		
phoneLock.authorized.x.value	**string**	
The number or address for an authorized contact		
Up to five (x=1 to 5) authorized contacts that a user can call while their phone is locked. Each contact needs a description to display on the screen, and a phone number or address value for the phone to dial.		
phoneLock.browserEnabled	**0 or 1**	**0**
If 0, the microbrowser or browser is not displayed while the phone is locked. If 1, the microbrowser or browser is displayed while the phone is locked.		
phoneLock.dndWhenLocked	**0 or 1**	**0**

Phone Lock Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.

<powerSaving/>

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.cecEnable	**0 or 1**	**0**
If 0, the RealPresence Trio Visual+ display behavior is controlled only by the value set for `powerSaving.tvStandbyMode`. If 1, when the RealPresence Trio 8800 enters power-saving mode, the RealPresence Trio Visual+ display switches to standby mode and powers up when the RealPresence Trio 8800 exits power-saving mode.		
powerSaving.tvStandbyMode	**noSignal, black**	**black**
Choose whether power-saving mode places a black screen on the RealPresence Trio Visual+ display or turns off the HDMI signal going to the RealPresence Trio Visual+ display.		

Power Saving Parameters (continued)

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** **RealPresence Trio=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. For RealPresence Trio, the default value is 1. When the RealPresence Trio the Base Profile is set to SkypeUSB, the default is 0. 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** **RealPresence Trio=30**
The number of minutes to wait while the phone is idle during office hours before activating power saving. Note that the default time varies by device model.		
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	**0 to 24**	**12**
powerSaving.officeHours.duration.Tuesday	**0 to 24**	**12**
powerSaving.officeHours.duration.Wednesday	**0 to 24**	**12**
powerSaving.officeHours.duration.Thursday	**0 to 24**	**12**
powerSaving.officeHours.duration.Friday	**0 to 24**	**12**
powerSaving.officeHours.duration.Saturday	**0 to 24**	**0**
powerSaving.officeHours.duration.Sunday	**0 to 24**	**0**
The duration of the day's office hours.		
powerSaving.officeHours.startHour.xxx	**0 to 23**	**7**
The starting hour for the day's office hours, where xxx is one of `monday, tuesday, wednesday, thursday, friday, saturday,` and `sunday` (refer to `powerSaving.officeHours.duration` for an example).		
powerSaving.userDetectionSensitivity.offHours	**0 to 10**	**2**

Power Saving Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.		

<pres/>

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.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.		

<prov/>

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.eula.accepted	**0 or 1**	**0**
If 0, the product End User License Agreement (EULA) that displays on the RealPresence Trio 8800 at initial startup must be manually accepted on each system. If 1, the EULA is accepted automatically for all systems.		
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.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**	**0**
If 0, the phone does not displays the software update notification and options, and the phone reboots automatically to update the software. If 1, the phone displays the software update notification and options, and the user can control the software download.		
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.		

[1] Change causes phone to restart or reboot.

<ptt/>

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.allowOffHookPages	**0 or 1**	**0**
If 0, PTT messages do not play out on the phone during an active call and the user must accept incoming PTT messages to play out. Priority and emergency PTT messages. If 1, incoming PTT messages play out even when there is an active call on the phone.		
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.channel.x.allowReceive	**0 or 1**	**1**
If 1, the phone receives incoming PTT messages on channel x. If 0, the channel does not receive incoming PTT messages.		
ptt.channel.x.allowTransmit	**0 or 1**	**1**
If 1, outgoing PTT messages are allowed on channel x. If 0, outgoing PTT messages are not allowed on channel x		
ptt.channel.x.available	**0 or 1**	**1**
If 1, channel x is available.If 0, channel x is not available.		
ptt.channel.x.label	**string**	**NULL**
Specify a label for channel x.		
ptt.channel.x.subscribed	**1 - 25**	**Channels 1, 24, 25 - 1 (default)** **Channels 2 - 23 - 0 (default)**
If 1, the PTT is subscribed for channel x. If 0, the PPT is not subscribed for channel x.		
ptt.codec	**G.711Mu, G.726QI, G.722**	**G.722**
Specify codec to use for PTT.		
ptt.defaultChannel	**1 - 25**	**1**
Specify the default channel number used for PTT transmissions.		
ptt.emergencyChannel site.cfg	**1 - 25**	**25**
Specify the channel to use for emergency PTT transmissions.		
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.		

Push-To-Talk and Group Paging Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.		
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.		

Push-To-Talk and Group Paging Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.		
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.		
ptt.payloadSize	**10, 20, 30, 40, 50, 60, 70, 80**	**20**
Specify the payload size for PTT transmissions.		
ptt.priorityChannel	**1 - 25**	**24**
Specify the channel number to use for priority PTT transmissions.		
ptt.pttMode.enable	**0 or 1**	**0**
If 0, PTT is disabled. If 1, PTT is enabled.		
ptt.volume	**-57 to 0**	**-20**
Controls the volume level for pages without changing the volume level for incoming calls.		

<qos/>

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.ethernet.tcpQosEnabled	**0 or 1**	**0**
0 (default) -- The phone does not send configured QoS priorities for SIP over TCP transport. 1 - The phone sends configured QoS priorities for SIP over TCP transport.		
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] **qos.ip.callControl.max_throughput[1]** **qos.ip.callControl.min_cost[1]** **qos.ip.callControl.min_delay[1]** **qos.ip.callControl.precedence[1]**	**0 or 1** **0 or 1** **0 or 1** **0 or 1** **0 -7**	**0** **0** **0** **1** **5**
Set the bits in the IP ToS field of the IP header used for call control. Specify whether or not to set the max reliability bit, the max throughput bit, the min cost bit, the min delay bit, and the precedence bits. If 0, the bit in the IP ToS field of the IP header is not set. If 1, the bit is set.		
qos.ip.rtp.dscp[1]	**0 to 63 or EF or any of AF11,AF12, AF13,AF21, AF22,AF23, AF31,AF32, AF33,AF41, AF42,AF43**	**Null**
Specify the DSCP of packets. If the value is not null, this parameter overrides the other `qos.ip.rtp.*` parameters. The default value is Null, so the other `quality.ip.rtp.*` parameters are used.		
qos.ip.rtp.max_reliability[1] **qos.ip.rtp.max_throughput[1]** **qos.ip.rtp.min_cost[1]** **qos.ip.rtp.min_delay[1]** **qos.ip.rtp.precedence[1]**	**0 or 1** **0 or 1** **0 or 1** **0 or 1** **0 -7**	**0** **1** **0** **1** **5**

Quality of Service (Type-of-Service) Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.		
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.		
qos.ip.rtp.video.max_reliability[1]	**0 or 1**	**0**
qos.ip.rtp.video.max_throughput[1]	**0 or 1**	**1**
qos.ip.rtp.video.min_cost[1]	**0 or 1**	**0**
qos.ip.rtp.video.min_delay[1]	**0 or 1**	**1**
qos.ip.rtp.video.precedence[1]	**0 -7**	**5**
Set the bits in the IP ToS field of the IP header used for RTP video. Specify whether or not to set the max reliability bit, the max throughput bit, the min cost bit, the min delay bit, and the precedence bit. If 0, the bit in the IP ToS field of the IP header is not set. If 1, the bit is set.		

[1] Change causes phone to restart or reboot.

<reg/>

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.		
reg.x.auth.domain	**string**	**Null**

Registration Parameters (continued)

Parameter	*Permitted Values*	*Default*
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**
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.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 as the default caller ID.		

Registration Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.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 r`eg.1.useteluriAsLineLabel=1`, then the tel URI/phone number/address displays as the label. • If r`eg.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**

Registration Parameters (continued)

Parameter	*Permitted Values*	*Default*
The line extension for a shared line. This parameter applies to private lines and BroadSoft call park and retrieve. If there is no extension provided for this parameter, the call park notification is ignored for the shared line.		
reg.x.lineKeys	**1 to max**	**1**
Specify the number of line keys to use for a single registration. The maximum number of line keys you can use per registration depends on your phone model.		
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.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**
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**

Registration Parameters (continued)

Parameter	*Permitted Values*	*Default*
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**	**1**
If 1, negotiation of SRTP compliant with Microsoft Session Description Protocol Version 2.0 Extensions is supported. This parameter overrides `sec.srtp.simplifiedBestEffort`. If 0, no SRTP is supported.		
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.		
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.		

Registration Parameters (continued)

Parameter	*Permitted Values*	*Default*
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**	**0**
Enables and disables all line keys.		
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**
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**

Registration Server Parameters (continued)

Parameter	*Permitted Values*	*Default*
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`.		
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**

Registration Server Parameters (continued)

Parameter	*Permitted Values*	*Default*
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 r`eg.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.		
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.		

<request/>

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.

<saf/>

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

<se/>

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 <pat/>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.		

<pat/>

Patterns use a simple script language that allows different chord sets or wave files to be strung together with periods of silence. The script language uses the instructions shown in the 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	**1-255**
Found in region.cfg	
se.pat.callProg.secondaryDialTone.inst.1.type	**0-255**
Found in region.cfg	
se.pat.callProg.secondaryDialTone.inst.1.value	**0-50**
Found in region.cfg	
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	**UTF-8 encoded string**
Sound effects name, where cat is `callProg`, `ringer`, or `misc`.	
se.pat.cat.x.inst.y.type	**sampled, chord, silence, branch**
Type of sound effect, where cat is `callProg`, `ringer`, or `misc`.	
se.pat.cat.x.inst.y.value	**String**
The instruction: `sampled` – sampled audio file number, `chord` – type of sound effect, `silence` – silence duration in ms, `branch` – number of instructions to advance. `cat` is `callProg`, `ringer`, or `misc`.	

The 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*
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
ringer12	Ringback-style	Short double ringback

Ringtone Pattern Names (continued)

Parameter Name	Ringtone Name	Description
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.		

<sec/>

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:

- <encryption/>
- <pwd/><length/>
- <srtp/>
- <H235/>
- <dot1x><eapollogoff/>
- <hostmovedetect/>

- <TLS/>

<encryption/>

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/><length/>

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.

<srtp/>

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**	**Skype = 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**	**1**

SRTP Parameters (continued)

Parameter	*Permitted values*	*Defaults*
If 1, negotiation of SRTP compliant with Microsoft Session Description Protocol Version 2.0 Extensions is supported. If 0, no SRTP 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><eapollogoff/>

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.

<hostmovedetect/>

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:

- <profile/>
- <profileSelection/>.

TLS Parameters

Parameter	*Permitted Values*	*Default*
sec.TLS.browser.cipherList	**String**	**NoCipher**
The cipher list for browser. The format for the cipher list uses OpenSSL syntax found at: http://www.openssl.org/docs/apps/ciphers.html.		
sec.TLS.cipherList	**String (1 - 1024 characters)**	**ALL:!aNULL:!eNULL:!DSS:!SEED:!ECDSA:!IDEA:!MEDIUM:!LOW:!EXP:!ADH:!ECDH:!PSK:!MD5:!RC4:@STRENGTH**

TLS Parameters (continued)

Parameter	*Permitted Values*	*Default*
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**
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.profileSelection.SOPI	**1 - 7**	**PlatformProfile1**
Select the platform profile you want to use. You can choose platform profile 1 - 7.		
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.		
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.		

TLS Parameters (continued)

Parameter	*Permitted Values*	*Default*
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[1]	**String (1 - 1024 characters)**	**ALL:!aNULL:!eNULL:!DSS:!SEED:!ECDSA:!IDEA:!MEDIUM:!LOW:!EXP:!ADH:!ECDH:!PSK:!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).		

TLS Profile Parameters (continued)

Parameter	*Permitted Values*	*Default*
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).		

<profileSelection/>

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.		

TLS Profile Selection Parameters (continued)

Parameter	*Permitted Values*	*Default*
sec.TLS.profileSelection.syslog	**PlatformProfile1 or PlatformProfile2**	**PlatformProfile1**
The TLS platform profile to use for syslog operations.		

<tcpIpApp/>

This parameter includes:

- <dhcp/>
- <dns/>
- <ice/>
- <sntp/>
- <port/><rtp/>
- <keepalive/>
- <fileTransfer/>

<dhcp/>

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.		

<dns/>

The <dns/> 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 <device/> 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.

<ice/>

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.		

<sntp/>

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/><rtp/>

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 SDPprotocols. If set to 1, the phone rejects RTP packets that arrive from non-negotiated IP addresses.		
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** **Skype = 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.

<keepalive/>

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.		
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.		
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.		
tcpIpApp.keepalive.tcp.sip.tls.enable	**0 or 1**	**0**
If 0, disable TCP keep-alive for SIP signaling connections that use TLS transport. If 1, enable TCP keep-alive for SIP signaling connections that use TLS transport.		

[1] Change causes phone to restart or reboot.

<fileTransfer/>

The parameters listed in the 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.		

<tones/>

This parameter lists configuration items for available tone resources and includes:

- <DTMF/>
- <chord/>

<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**	**Skype = 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/>

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 **tone.chord.misc.x.freq.y** **tone.chord.ringer.x.freq.y**	**0-1600** **0-1600** **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 **tone.chord.misc.x.level.y** **tone.chord.ringer.x.level.y**	**-57 to 3** **-57 to 3** **-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 **tone.chord.misc.x.onDur** **tone.chord.ringer.x.onDur**	**positive integer** **positive integer** **positive integer**
The on duration (length of time to play each component) in milliseconds, 0=infinite.	
tone.chord.callProg.x.offDur **tone.chord.misc.x.offDur** **tone.chord.ringer.x.offDur**	**positive integer** **positive integer** **positive integer**
The off duration (the length of silence between each chord component) in milliseconds, 0=infinite.	
tone.chord.callProg.x.repeat **tone.chord.misc.x.repeat** **tone.chord.ringer.x.repeat**	**positive integer** **positive integer** **positive integer**
The number of times each ON/OFF cadence is repeated, 0=infinite.	

<up/>

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.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.		
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.		

User Preferences Parameters (continued)

Parameter	*Permitted Values*	*Default*
up.formatPhoneNumbers	**0 or 1**	**1**
Enable or disable automatic number formatting.		
up.handsfreeMode	**0 or 1**	**1**
If 0, the speakerphone is disabled (cannot be used). If 1, the speakerphone is enabled.		
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.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**

User Preferences Parameters (continued)

Parameter	*Permitted Values*	*Default*
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=3** **VVX 600/601=4** **RealPresence Trio=3**
Set the maximum number of columns the VVX 500/501, 600/601, or RealPresence Trio solution 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.oneTouchDirectory	**0 or 1**	**1**
Enable or disable display of the Address Book icon on the main menu and the Skype for Business Directory search option.		
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` .		
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 - 16**	**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. 1-16 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**

User Preferences Parameters (continued)

Parameter	*Permitted Values*	*Default*
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**
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**

User Preferences Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.

<upgrade/>

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, and RealPresence Trio solution.

This parameter also includes:

- <camera/>
- <codecs/>

- <localCameraView/>

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.callRate	**128 to 2048**		**512**
The default call rate (in kbps) to use when initially negotiating bandwidth for a video call.			
video.conf.profile	540p 1080p 720p 360p 240p 180p		**540p**
Set the video resolution for the large video window in all layouts.			

Video Parameters (continued)

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, 1**	**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.mute.sendCannedVideo	**0 or 1**	**1**
1 (default) - The RealPresence Trio system sends a custom image to the far end when you press Stop my video. 0 - The RealPresence Trio system sends no video to the far end when you press Stop my video. A default no video graphic displays.		
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.

<camera/>

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).		

<codecs/>

The video codecs include:

- <codecPref/>
- <profile/>

<codecPref/>

The following table lists video codec parameters and specifies the video codec preferences for the RealPresence Trio solution. To disable codecs, set the value to 0. A value of 1 indicates the codec is the most preferred and has highest priority.

Video Codec Preference Parameters

Parameter	*Permitted Values*	*Default*
video.codecPref.H264	**0 - 8**	**4**
video.codecPref.H264HP	**0 - 8**	**2**
Set the H.264 High Profile video codec preference priority.		
video.codecPref.H264HP.packetizationMode0	**0 - 8**	**5**
video.codecPref.H264SVC		
video.codecPref.Xdata	**0 - 8**	**7**
Set the Remote Desktop Protocol (RDP) codec preference priority. 1 indicates the codec is the most preferred and has highest priority.		
video.codecPref.XH264UC	**0 - 8**	**1**
Set the Microsoft H.264 UC video codec preference priority.		
video.codecPref.XUlpFecUC	**0 - 8**	**8**
Set the forward error correction (FEC) codec priority.		

<profile/>

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.		
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’.		
video.profile.H261.jitterBufferMax[1]	**(video.profile.H261.jitterBufferMin + 500ms) to 2500ms**	**2000ms**

Video Profile Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.		
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.		
video.profile.H261.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.H261.payloadType[1]	**0 to 127**	**31**
RTP payload format type for H261 MIME type.		
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.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.H263.payloadType[1]	**0 to 127**	**34**
RTP payload format type for H263 MIME type.		

Video Profile Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.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.jitterBufferMax[1]	**(video.profile.H2631998.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.H2631998.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 Parameters (continued)

Parameter	*Permitted Values*	*Default*
video.profile.H2631998.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.H2631998.payloadType[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.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.		
video.profile.H264HP.jitterBufferMax	**533 - 2500 milliseconds**	**2000**

Video Profile Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.H264HP.jitterBufferMin	**33 - 1000 milliseconds**	**150**
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.H264HP.jitterBufferShrink	**33 - 1000**	**70**
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.H264HP.payloadType	**0 - 127**	**100**
RTP payload format type for H264/90000 MIME type.		
video.profile.H264HP.profileLevel	**String (1 - 5 characters)**	**4.1**
Specify the highest profile level within the baseline profile supported in video calls.		
video.profile.Xdata.payloadType	**0 - 127**	**127**
Specify the payload type to use in SDP negotiations of the payload used for Skype for Business desktop content sharing.		
video.profile.XH264UC.jitterBufferMax	**533 - 2500 milliseconds**	**2000**
The largest jitter buffer depth to support. 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.XH264UC.jitterBufferMin	**33 - 1000 milliseconds**	**150**
The smallest jitter buffer depth that must be achieved before playout begins for the first time. After this depth has been achieved initially, the depth may fall below this point and playout 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.XH264UC.jitterBufferShrink	**33 - 1000**	**70**
The minimum duration in milliseconds of Real-time Transport Protocol (RTP) packet Rx with no packet loss that will trigger jitter buffer size shrinks. Use smaller values (1000 ms) to minimize the delay on known good networks.		
video.profile.XH264UC.mstMode	**String**	**NI-TC**

Video Profile Parameters (continued)

Parameter	*Permitted Values*	*Default*
Specify the multi-session transmission packetization mode. The value of NI-TC identifies non-interleaved combined timestamp and CS-DON mode. This value should not be modified for interoperation with other Skype for Business devices.		
video.profile.XH264UC.payloadType	**0 - 127**	**122**
RTP payload format type for H.264 MIME type.		
video.profile.XUlpFecUC.alwaysOn	**0 or 1**	**1**
video.profile.XUlpFecUC.debug.rxDropBurst	**1 - 100**	**1**
video.profile.XUlpFecUC.debug.rxDropOnlyLayer0	**0 or 1**	**1**
video.profile.XUlpFecUC.debug.rxDropRate	**0 - 40000**	**0**
video.profile.XUlpFecUC.debug.txDropBurst	**1 - 100**	**1**
video.profile.XUlpFecUC.debug.txDropRate	**0 - 40000**	**0**
video.profile.XUlpFecUC.noLossTurnOffTimeout	**10 - 7200**	**300**
video.profile.XUlpFecUC.payloadType	**0 - 127**	**123**
video.profile.XUlpFecUC.rxEnabled	**0 or 1**	**1**
video.profile.XUlpFecUC.txEnabled	**0 or 1**	**1**

Video Profile Parameters (continued)

Parameter	*Permitted Values*	*Default*
video.simpleJB.enable	**0 or 1**	**1**
video.simpleJB.lipSyncDelayMs	**0 - 250 ms**	**0**
video.simpleJB.timeoutMs	**0 - 250 ms**	**100 ms**
video.rtcpbandwidthdetect.enable	**0 or 1**	**0**
If 1, RealPresence Trio 8800 uses an estimated bandwidth value from the RTCP message to control Tx/Rx video bps.		

1 Change causes phone to restart or reboot.

<localCameraView/>

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.		

<voice/>

The parameters listed in the following tables configure phone audio.

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**
If 0, no Tx filtering is performed. If 1, narrowband Tx high pass filter is enabled.		

[1] Change causes phone to restart or reboot.

<aec/>

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.		

<aes/>

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.		

<cn/>

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.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**

(continued)Comfort Noise Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.		

<codecPref/>

As of Polycom UC Software 3.3.0, you can configure a simplified set of codec properties for all phone models to improve consistency and reduce workload on the phones. Phone codec preferences are listed in the 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.		

<ns/>

This section lists noise suppression parameters available with the Polycom NoiseBlock feature.

<vad/>

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**

Voice Activity Detection (VAD) Parameters (continued)

Parameter	*Permitted Values*	*Default*
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**
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.

<quality monitoring/>

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 Quality Monitoring Parameters (continued)

Parameter	*Permitted Values*	*Default*
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**
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 Quality Monitoring Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.

<rxQoS/>

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.		
voice.rxQos.mr.avgJitter	**0-200**	**10**
voice.rxQos.mr.maxJitter	**20-500**	**30**
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**

Voice Jitter Buffer Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.

<voIpProt/>

You must set up the call server and DTMF signaling parameters.

This parameter includes:

- <server/>
- <SDP/>
- <SIP/>

<server/>

The next table describes VoIP server configuration parameters.

VoIP Server Parameters

Parameter	*Permitted Values*	*Default*
voIpProt.server.dhcp.available[1]	**0 or 1**	**0**
If 0, do not check with the DHCP server for the SIP server IP address. If 1, check with the server for the IP address.		
voIpProt.server.dhcp.option[1]	**128 to 254**	**128**
The option to request from the DHCP server if `voIpProt.server.dhcp.available`= 1. Note: If `reg.x.server.y.address` is non-Null, it takes precedence even if the DHCP server is available.		
voIpProt.server.dhcp.type[1]	**0 or 1**	**0**
Type to request from the DHCP server if `voIpProt.server.dhcp.available` is set to 1.If this parameter is set to 0, IP request address. If set to 1, request string		
voIpProt.server.x.address	**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.		

VoIP Server Parameters (continued)

Parameter	*Permitted Values*	*Default*
voIpProt.server.x.expires	**positive integer, minimum 10**	**3600**
The phone's requested registration period in seconds. Note: The period negotiated with the server may be different. The phone will attempt to re-register at the beginning of the overlap period. For example, if expires="300" and overlap="5", the phone will re-register after 295 seconds (300-5).		
voIpProt.server.x.expires.lineSeize	**positive integer, minimum 10**	**30**
Requested line-seize subscription period.		
voIpProt.server.x.expires.overlap	**5 to 65535**	**60**
The number of seconds before the expiration time returned by server x at which the phone should try to re-register. The phone will try to re-register at half the expiration time returned by the server if the server value is less than the configured overlap value.		
voIpProt.server.x.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`.		
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**

VoIP Server Parameters (continued)

Parameter	*Permitted Values*	*Default*
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**
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.		
voIpProt.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.		
voIpProt.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.		
voIpProt.server.x.transport	**DNSnaptr, TCPpreferred, UDPOnly, TLS, TCPOnly**	**DNSnaptr**

VoIP Server Parameters (continued)

Parameter	*Permitted Values*	*Default*
The transport method the phone uses to communicate with the SIP server. • **Null** or **DNSnaptr** If `voIpProt.server.x.address` is a hostname and `voIpProt.server.x.port` is 0 or Null, do NAPTR then SRV look-ups to try to discover the transport, ports and servers, as per RFC 3263. If `voIpProt.server.x.address` is an IP address, or a port is given, then UDP is used. • **TCPpreferred** TCP is the preferred transport; UDP is used if TCP fails. • **UDPOnly** Only UDP will be used. • **TLS** If TLS fails, transport fails. Leave port field empty (will default to 5061) or set to 5061. • **TCPOnly** Only TCP will be used.		
voIpProt.server.x.protocol.SIP	**0 or 1**	**1**
If 1, server is a SIP proxy/registrar. Note: if set to 0, and the server is confirmed to be a SIP server, then the value is assumed to be 1.		
voIpProt.server.x.expires	**positive integer, minimum 10**	**3600**
The phone's requested registration period in seconds. Note: The period negotiated with the server may be different. The phone will attempt to re-register at the beginning of the `overlap` period. For example, if `expires="300"` and `overlap="5"`, the phone will re-register after 295 seconds (300–5).		
voIpProt.server.x.expires.overlap	**5 to 65535**	**60**
The number of seconds before the expiration time returned by server x at which the phone should try to re-register. The phone will try to re-register at half the expiration time returned by the server if the server value is less than the configured overlap value.		
voIpProt.server.x.expires.lineSeize	**positive integer, minimum 0 was 10**	**30**
Requested line-seize subscription period.		
voIpProt.server.x.failOver.failBack.mode	**newRequests, DNSTTL, registration, duration**	**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**

VoIP Server Parameters (continued)

Parameter	*Permitted Values*	*Default*
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.		
voIpProt.server.x.retryMaxCount	**0 to 20**	**3**
If set to 0, 3 is used. The number of retries that will be attempted before moving to the next available server.		
voIpProt.server.x.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.		
voIpProt.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.		
voIpProt.server.x.useOutboundProxy	**0 or 1**	**1**
Specify whether or not to use the outbound proxy specified in `voIpProt.SIP.outboundProxy.address` for server x.		

[1] Change causes phone to restart or reboot.

<SDP/>

The 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.		
voIpProt.SDP.answer.useLocalPreferences.video	**0 or 1**	**1**
1 (default) - The phone uses its own preference list instead of the preference list in the offer when deciding which video codec to use. 0 - The phone's use of its own preference list is disabled. Allows you to reset the parameter `voIpProt.SDP.answer.useLocalPreferences` to the default 0 for audio only.		
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.		
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.)		
voIpProt.SIP.alertInfo.x.class	**see the list of ring classes in <rt/>**	**default**
Alert-Info fields from INVITE requests will be compared against as many of these parameters as are specified (x=1, 2, ..., N) and if a match is found, the behavior described in the corresponding ring class is applied.		
voIpProt.SIP.alertInfo.x.value	**string**	**Null**
A string to match the Alert-Info header in the incoming INVITE.		
voIpProt.SIP.allowTransferOnProceeding	**0, 1, 2**	**1**
If set to 0, a transfer is not allowed during the proceeding state of a consultation call. If set to 1, a transfer can be completed during the proceeding state of a consultation call. If set to 2, phones will accept an INVITE with replaces for a dialog in early state. This is needed when using transfer on proceeding with a proxying call server such as openSIPS, reSIProcate or SipXecs.		
voIpProt.SIP.authOptimizedInFailover	**0 or 1**	**0**
If set to 1, when failover occurs, the first new SIP request is sent to the server that sent the proxy authentication request. If set to 0, when failover occurs, the first new SIP request is sent to the server with the highest priority in the server list. If `reg.x.auth.optimizedInFailover` set to 0, this parameter is checked. If `voIpProt.SIP.authOptimizedInFailover` is 0, then this feature is disabled. If both parameters are set, the value of `reg.x.auth.optimizedInFailover` takes precedence.		
voIpProt.SIP.callinfo.precedence.overAlertinfo	**0 or 1**	**0**
Give priority to call-info header with answer-after string over alert-info.		
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.		

Session Initiation Protocol (SIP) Parameters (continued)

Parameter	Permitted Values	Default
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`.		
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.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.		

Session Initiation Protocol (SIP) Parameters (continued)

Parameter	Permitted Values	Default
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.		
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.		
voIpProt.SIP.keepalive.sessionTimers	**0 or 1**	**0**
If set to 1, the session timer will be enabled. If set to 0, the session timer will be disabled, and the phone will not declare “timer” in “Support” header in an INVITE. The phone will still respond to a re-INVITE or UPDATE. The phone will not try to re-INVITE or UPDATE even if the remote endpoint asks for it.		
voIpProt.SIP.lineSeize.retries	**3 to 10**	**10**
Controls the number of times the phone will retry a notify when attempting to seize a line (BLA).		
voIpProt.SIP.local.port[1]	**0 to 65535**	**5060**
The local port for sending and receiving SIP signaling packets. If set to 0, 5060 is used for the local port but is not advertised in the SIP signaling. If set to some other value, that value is used for the local port and it is advertised in the SIP signaling.		
voIpProt.SIP.looseContact	**0 or 1**	**0**
0 (default) - The port parameter is added to the contact header in TLS case. 1 - The port parameter is not added to the contact header or SIP messages.		
voIpProt.SIP.ms-forking	**0 or 1**	**0**

Session Initiation Protocol (SIP) Parameters (continued)

Parameter	Permitted Values	Default
If set to 0, support for MS-forking is disabled. If set to 1, support for MS-forking is enabled and the phone will reject all Instant Message INVITEs. This parameter is applies when installing Microsoft Live Communications Server. Note that if any endpoint registered to the same account has MS-forking disabled, all other endpoints default back to non-forking mode. Windows Messenger does not use MS-forking so be aware of this behavior if one of the endpoints is using Windows Messenger.		
voIpProt.SIP.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`.		
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.		
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.		
voIpProt.SIP.outboundProxy.port	**0 to 65535**	**0**
The port of the SIP server to which the phone sends all requests.		
voIpProt.SIP.outboundProxy.failOver.failBack.mode	**newRequests, DNSTTL, registration, duration,**	**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.		
voIpProt.SIP.outboundProxy.failOver.failBack.timeout	**0, 60 to 65535**	**3600**
The time to wait (in seconds) before failback occurs (overrides `voIpProt.server.x.failOver.failBack.timeout`).If the fail back mode is set to Duration, the phone waits this long after connecting to the current working server before selecting the primary server again. If 0, the phone will not fail-back until a fail-over event occurs with the current server.		
voIpProt.SIP.outboundProxy.failOver.failRegistrationOn	**0 or 1**	**0**
When set to 1, and the reRegisterOn parameter is enabled, the phone will silently invalidate an existing registration (if it exists), at the point of failing over. When set to 0, and the reRegisterOn parameter is enabled, existing registrations will remain active. This means that the phone will attempt failback without first attempting to register with the primary server to determine if it has recovered. Note that `voIpProt.SIP.outboundProxy.failOver.RegisterOn` must be enabled.		
voIpProt.SIP.outboundProxy.failOver.onlySignalWithRegistered	**0 or 1**	**1**

Session Initiation Protocol (SIP) Parameters (continued)

Parameter	Permitted Values	Default
When set to 1, and the reRegisterOn and failRegistrationOn parameters are enabled, no signaling is accepted from or sent to a server that has failed until failback is attempted or failover occurs. If the phone attempts to send signaling associated with an existing call via an unregistered server (for example, to resume or hold a call), the call will end. No SIP messages will be sent to the unregistered server. When set to 0, and the reRegisterOn and failRegistrationOn parameters are enabled, signaling will be accepted from and sent to a server that has failed (even though failback hasn't been attempted or failover hasn't occurred). This parameter overrides `voIpProt.server.x.failOver.onlySignalWithRegistered`.		
voIpProt.SIP.outboundProxy.failOver.reRegisterOn	**0 or 1**	**0**
This parameter overrides the `voIpProt.server.x.failOver.reRegisterOn`. When set to 1, the phone will attempt to register with (or via, for the outbound proxy scenario), the secondary server. If the registration succeeds (a 200 OK response with valid expires), signaling will proceed with the secondary server. When set to 0, the phone won't attempt to register with the secondary server, since the phone will assume that the primary and secondary servers share registration information.		
voIpProt.SIP.outboundProxy.transport	**DNSnaptr, TCPpreferred, UDPOnly, TLS, TCPOnly**	**DNSnaptr**
The transport method the phone uses to communicate with the SIP server. • **Null** or **DNSnaptr** If `reg.x.outboundProxy.address` is a hostname and `reg.x.outboundProxy.port` is 0 or Null, do NAPTR then SRV look-ups to try to discover the transport, ports and servers, as per RFC 3263. If `reg.x.outboundProxy.address` is an IP address, or a port is given, then UDP is used. • **TCPpreferred** TCP is the preferred transport, UDP is used if TCP fails. • **UDPOnly** Only UDP will be used. • **TLS** If TLS fails, transport fails. Leave port field empty (will default to 5061) or set to 5061. • **TCPOnly** Only TCP will be used.		
voIpProt.SIP.pingInterval	**0 to 3600**	**0**
The number in seconds to send PING message. This feature is disabled by default.		
voIpProt.SIP.pingMethod	**PING, OPTIONS**	**PING**
The ping method to be used.		
voIpProt.SIP.presence.nortelShortMode[1]	**0 or 1**	**0**
Different headers sent in SUBSCRIBE when used for presence on an Avaya (Nortel) server. Support is indicated by adding a header `Accept-Encoding: x-nortel-short`. A PUBLISH is sent to indicate the status of the phone.		
voIpProt.SIP.requestValidation.digest.realm[1]	**A valid string**	**PolycomSPIP**
Determines the string used for Realm.		
voIpProt.SIP.requestValidation.x.method[1]	**Null, source, digest, both, all**	**Null**

Session Initiation Protocol (SIP) Parameters (continued)

Parameter	Permitted Values	Default
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.		
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).		

Session Initiation Protocol (SIP) Parameters (continued)

Parameter	Permitted Values	Default
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**
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.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**

Session Initiation Protocol (SIP) Parameters (continued)

Parameter	Permitted Values	Default
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**
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.		
voipProt.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.		

Session Initiation Protocol (SIP) Parameters (continued)

Parameter	Permitted Values	Default
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**
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.

<webutility/>

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.		

<xmpp/>

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.		

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.

The Updater, UC Software executable, and configuration files can all be updated using centralized provisioning.

Update Software with a USB Flash Drive

You can use an USB flash drive to update the software on the RealPresence Trio solution or to provision and configure the system.

When you configure the system using a USB drive, the configuration on the USB overrides all previous configurations. However, when the USB drive is removed, the system returns to the previous configuration.

To update or provision the RealPresence Trio 8800 using an USB flash drive:

1. Format a USB flash drive as FAT32. Polycom recommends that you use a USB 2.0 flash drive.

   If you are using a drive that is already formatted, ensure that previous files are deleted from the flash drive.
2. From Polycom Voice Support, download the software package.
3. Place the 3111-65290-001.sip.ld file in the root directory of the flash drive. If provisioning the system, place the 000000000000.cfg or <MAC>.cfg file and any configuration files in the root directory as well.
4. Connect the USB flash drive to the USB port on the system.
5. Enter the administrator password.

   The system detects the flash drive and starts the update within 30 seconds. The mute keys' indicator lights begin to flash, indicating that the update has started.

   The system reboots several times during the update. The update is complete when the indicator lights stop flashing and the Home screen displays.

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	0 (default) - The phone does not display the software update notification and options and the phone reboots automatically to update the software. 1 - The phone displays the software update notification and options and the user can control the software download.
`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.

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
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.

Restart, Reset to Defaults, Upload Log Files

You can restart, reset to defaults, and upload log files from the phone menu.

Restart the RealPresence Trio Visual+ System

You can restart the Trio Visual+ connected to the RealPresence Trio 8800.

To restart the RealPresence Trio Visual+:

» On the RealPresence Trio 8800 system Home screen, go to **Settings > Basic > Restart Networked Devices**.

Restart the RealPresence Trio 8800 and Visual+ Systems

You can restart the RealPresence Trio 8800 and Visual+ together.

To restart the RealPresence Trio 8800 and Visual+:

» On the RealPresence Trio 8800 system Home screen, go to **Settings > Basic > Restart System**.

Reset the RealPresence Trio 8800 and Visual+ to Factory Defaults

You can reset the RealPresence Trio 8800 and Visual+ systems to factory default settings. Resetting to defaults clears the flash parameters, removes log files, user data, and cached data, and resets the administrator password to 456.

Reset the RealPresence Trio 8800 to Defaults

You can reset the RealPresence Trio solution to defaults at power up.

To reset RealPresence Trio 8800 to factory defaults at power up:

1. Power on the RealPresence Trio 8800.
2. When the Polycom logo shows on the screen, press and hold the four corners of the LCD display screen.

3 Let go when the Mute light begins flashing.

Reset the RealPresence Trio Visual+ to Defaults

You can reset the RealPresence Trio Visual+ to defaults from the interface at power up.

To reset the RealPresence Trio Visual+ to factory defaults at power up:

1. Power on the RealPresence Trio Visual+.
2. When the pairing button light turns on, press and hold the pair button.
3. Let go of the pair button when the light begins flashing.

Upload RealPresence Trio System Log Files

You can upload log files to your provisioning server. Uploading log files copies the log files from the phone to the provisioning server. and creates new files named ***<MACaddress>-now-xxx.log***.

To upload log files:

1. Go to **Settings > Advanced >** Enter the administrator password (default 456) **> Administration Settings > Upload Configuration**.
2. Select one or more sources to upload from:
   - All Sources
   - Configuration Files
   - Local
   - MR
   - Web
   - SIP
3. Press **Upload**.

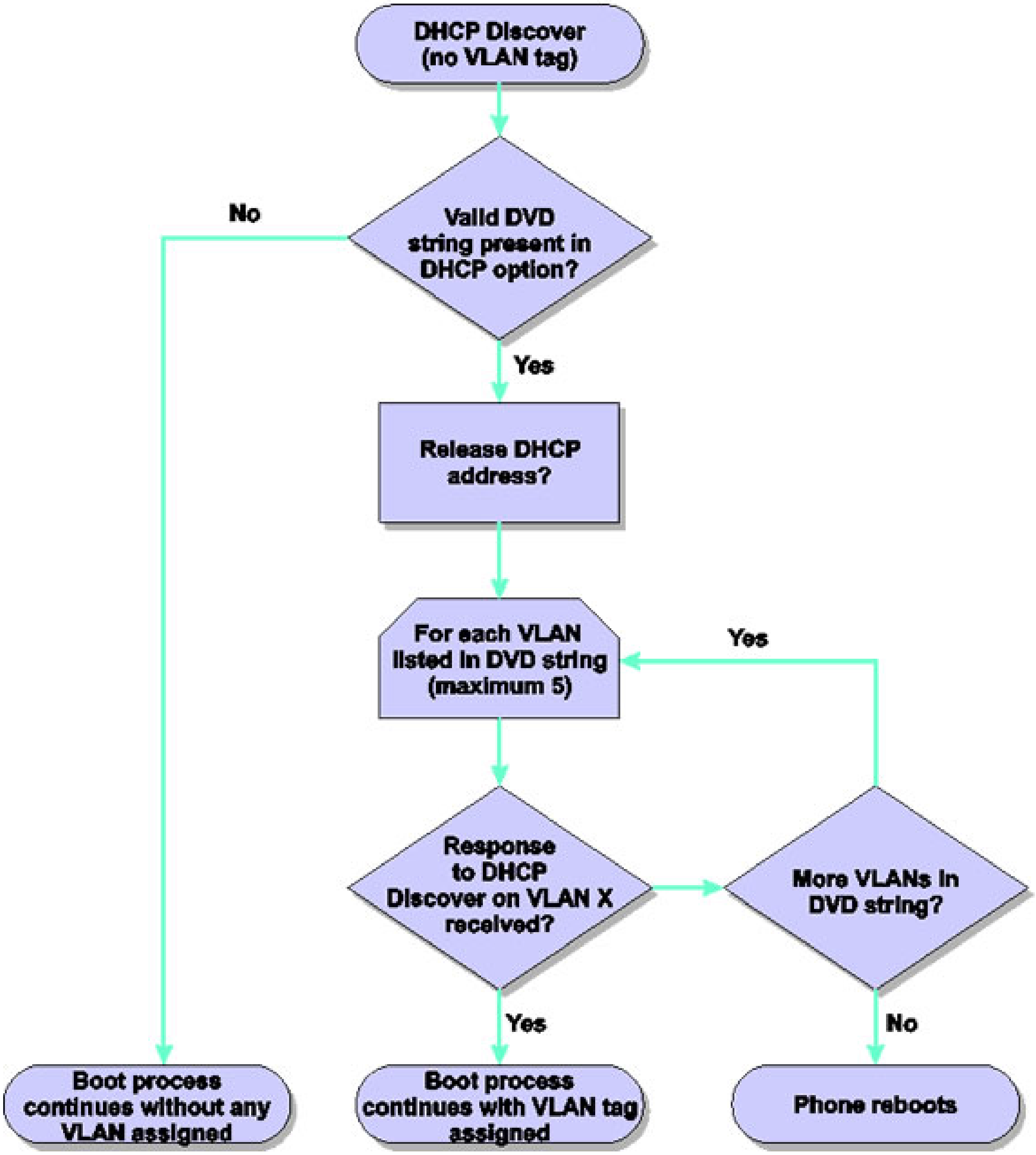
Assign a VLAN ID Using DHCP

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


DHCP Discover (no VLAN tag)
Valid DVD string present in DHCP option?
No
Yes
Release DHCP address?
For each VLAN listed in DVD string (maximum 5)
Yes
Response to DHCP Discover on VLAN X received?
More VLANs in DVD string?
Yes
No
Boot process continues without any VLAN assigned
Boot process continues with VLAN tag assigned
Phone reboots

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 the RealPresence Trio 8800 system.

- Sub-option 2 (part), length, "Real Presence Trio-Trio_8800"

```
02 1a 52 65 61 6c 50 72 65 73 65 6e 63 65 54 72 69 6f 2d 54 72 69 6f 5f 38 38 30 30
```

- Sub-option 3 (part number), length, "3111-65290-001,5"

```
03 10 33 31 31 31 2d 36 35 32 39 30 2d 30 30 31 2c 35
```

- Sub-option 4 (Application version), length, "SIP/5.4.1.16972/04-Jan-16 16:05"

```
05 1d 53 49 50 2f 35 2e 34 2e 31 2e 31 36 39 37 32 2f 30 34 2d 4a 61 6e 2d 31
36 20 31 36 3a 30 35
```

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 **Edit**.

3 Select **Disabled** and press **OK**.

4 Press **Exit** 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 the RealPresence Trio solution does not support expansion modules.

Before you can take a screen capture, you must provide power and connect the expansion module to a phone, and enable the phone’s web server using the parameter `httpd.enabled.`

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 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, PD Requested Power Value, PSE allocated 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**	**-**	**-**
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**	**-**	**-**	**-**

Supported TLVs

<table>
<tr><th>No</th><th>Name</th><th>Type (7 bits) [0-6]</th><th>Length (9 bits) [7-15]</th><th>Type Length</th><th>Org. Unique Code (3 bytes)</th><th>Sub Type</th></tr>
<tr><td colspan="7">Manufacturer’s name - “Polycom”; Hardware version; Application version; BootROM version</td></tr>
<tr><td><b>7</b></td><td><b>Capabilities</b></td><td><b>7</b></td><td><b>4</b></td><td><b>0x0e04</b></td><td><b>-</b></td><td><b>-</b></td></tr>
<tr><td colspan="7">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.</td></tr>
<tr><td><b>8</b></td><td><b>Management Address</b></td><td><b>8</b></td><td><b>12</b></td><td><b>0x100c</b></td><td><b>-</b></td><td><b>-</b></td></tr>
<tr><td colspan="7">Address String Len - 5, IPV4 subtype, IP address, Interface subtype - “Unknown”, Interface number - “0”, ODI string Len - “0”</td></tr>
<tr><td><b>9</b></td><td><b>IEEE 802.3 MAC/PHY config/status[1]</b></td><td><b>127</b></td><td><b>9</b></td><td><b>0xfe09</b></td><td><b>0x00120f</b></td><td><b>1</b></td></tr>
<tr><td colspan="7">Auto Negotiation Supported - “1”, enabled/disabled, Refer to PMD Advertise and Operational MAU.</td></tr>
<tr><td><b>10</b></td><td><b>LLDP-MED capabilities</b></td><td><b>127</b></td><td><b>7</b></td><td><b>0xfe07</b></td><td><b>0x0012bb</b></td><td><b>1</b></td></tr>
<tr><td colspan="7">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</td></tr>
<tr><td><b>11</b></td><td><b>LLDP-MED network policy[2]</b></td><td><b>127</b></td><td><b>8</b></td><td><b>0xfe08</b></td><td><b>0x0012bb</b></td><td><b>2</b></td></tr>
<tr><td colspan="7">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</td></tr>
<tr><td><b>12</b></td><td><b>LLDP-MED network policy[2]</b></td><td><b>127</b></td><td><b>8</b></td><td><b>0xfe08</b></td><td><b>0x0012bb</b></td><td><b>2</b></td></tr>
<tr><td colspan="7">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.
<b>Note</b>: Voice signaling TLV is sent only if it contains configuration parameters that are different from voice parameters.</td></tr>
<tr><td><b>13</b></td><td><b>LLDP-MED network policy[2]</b></td><td><b>127</b></td><td><b>8</b></td><td><b>0xfe08</b></td><td><b>0x0012bb</b></td><td><b>2</b></td></tr>
<tr><td colspan="7">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.</td></tr>
<tr><td><b>14</b></td><td><b>LLDP-MED location identification[3]</b></td><td><b>127</b></td><td><b>min len > 0, max len &lt;= 511</b></td><td><b>-</b></td><td><b>0x0012bb</b></td><td><b>3</b></td></tr>
</table>

Supported TLVs

No	Name	Type (7 bits) [0-6]	Length (9 bits) [7-15]	Type Length	Org. Unique Code (3 bytes)	Sub Type
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**	**12**	**0xfe07**	**0x00120F**	**4**
PowerType -PD device PowerSource-PSE&local Power Priority -Unknown, PD Requested Power Value depends on power configuration. If both PSE power and USB charging are disabled then it is 13W. Otherwise, it is 25.5W. This TLV is sent only by the RealPresence Trio 8800 system. The RealPresence Trio Visual+ relies on a hardware handshake only for power negotiations.						
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						
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
Trio 8800	Polycom RealPresence Trio 8800	RealPresenceTrio-Trio_8800
Trio Visual+	Polycom RealPresence Trio Visual+	RealPresenceTrio-Trio_Visual+

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)
Trio 8800	5.8	25.5
Trio Visual+	4.3	na

RealPresence Trio 8800 Power Management

Power available to the RealPresence Trio solution is limited and you must choose how to power the system and which features to enable or disable. You need to consider these powering options even if you are powering from a POE+ source.

USB Port Power Management

Device charging with the USB port on the RealPresence Trio 8800 system is disabled by default and when disabled the USB host port provides 100mA of power for peripheral devices. USB charging is disabled when powering the RealPresence Trio Visual+ from a LAN Out port.

To enable USB charging, you must power your RealPresence Trio 8800 system with an IEEE 802.3at Power over Ethernet **Plus** (PoE**+**) compliant power source. When USB charging is enabled, you can power and charge USB 2.0 compliant devices having a power draw of up to 1.500mA/7.5W.

Using Power over Ethernet (POE) Class 0

Powering the RealPresence Trio 8800 system from a Power over Ethernet (POE) Class 0 source provides full core functionality and results in the following limitations:

- The LAN Out port does not provide PoE power but otherwise is fully functional.

Using Power Sourcing Equipment Power (PoE PSE Power)

You can use Power Sourcing Equipment Power (PoE PSE Power) to power a RealPresence Trio Visual+ system from the LAN OUT port of the RealPresence Trio 8800 system.

To use PoE PSE Power, you must power the RealPresence Trio 8800 system with an IEEE 802.3at Power over Ethernet **Plus** (PoE**+**) compliant power source.

You cannot enable USB Charging of the USB host port and PSE PoE Power of LAN OUT port at the same time. If both are enabled, the RealPresence Trio system uses PSE PoE Power and ignores the USB charging setting.

Configure RealPresence Trio System Power Management

Use the parameters listed to manage the RealPresence Trio system’s power usage.

Parameter template	Permitted Values
`poe.pse.class`	Specify the LAN OUT PoE class. 0 (default) 0 - 3
`poe.pse.enabled`	1 (default) - The RealPresence Trio 8800 LAN OUT interface provides PoE power to a connected device. 0 - PoE power is not provided by the LAN OUT port.
`usb.charging.enabled`	0 (default) - You cannot charge USB-connected devices from the USB charging port. 1 - Enable fast charging of devices connected by USB port up to 7.5W power / 1.5A current.

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 menu. The location of the Warnings menu varies by model:

- RealPresence Trio 8800 **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
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.

Polycom UC Software Error Messages

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 microphones and speaker. • 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 > Warnings.**

2 Choose from these tests:

- ➢ **Audio Diagnostics** Test the speaker, microphone, handset, and a third party headset**.**
- ➢ **Display Diagnostics** Test the LCD for faulty pixels.
- ➢ **Touch Screen Diagnostics** Test the touch screen response.

Upload a Phone's Configuration

You can upload the phone's current configuration files from the phone menu 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.

You can use the Web Configuration Utility to upload the files.

To upload the phone’s current configuration:

1. Navigate to **Settings** > **Advanced** > **Admin Settings** > **Upload Configuration**.
2. Choose which files to upload: **All Sources**, **Configuration Files**, **Local**, **MR**, **Web**, or **SIP**. 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.

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.

When the RealPresence Trio Visual+ system is paired with a RealPresence Trio 8800, logging information from both devices is written to the same log files.

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**
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**

Managing Phone Memory Resources

Feature	Typical Memory Size
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.

Troubleshooting Power and Startup Issues

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.

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 phone does not ring.

Troubleshooting Calling Issues

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.
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.

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.

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 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.

Inbound and Outbound Ports for RealPresence Trio 8800 System

This section provides port usage information when configuring network equipment to support the RealPresence Trio system.

Inbound Ports for RealPresence Trio 8800 System

The following table lists the inbound IP ports currently used by the Polycom UC Software running on the RealPresence Trio 8800 system.

Inbound IP Port Connections to RealPresence Trio Systems

Inbound Port	*Type*	*Protocol*	*Function*	*Default*	*Configurable Port Number*
22	static	TCP	SSH Administration	Off	No
80	static	TCP	HTTP Pull Web interface, HTTP Push	Off	Yes

Inbound IP Port Connections to RealPresence Trio Systems

Inbound Port	*Type*	*Protocol*	*Function*	*Default*	*Configurable Port Number*
443	static	TCP	HTTP Pull Web interface, HTTP Push	On	Yes
1023	static	TCP	Telnet Diagnostics	Off	No
2222	Dynamic (2222 - 2269)	TCP/UDP	RTP media packets	On	Yes tcpIpApp.port.rtp.mediaPortRangeStart
2223	Dynamic (2222 - 2269)	TCP/UDP	RTCP media packets statistics	On	Yes tcpIpApp.port.rtp.mediaPortRangeStart
5001	static	TCP	People+Content IP	On	No
5060	static	TCP/UDP	SIP signaling	On	No
5061	static	TLS	SIP over TLS signaling	On	No
8001	static	TCP	HTTPS for modular room provisioning	On	Yes mr.deviceMgmt.port

Outbound Ports for RealPresence Trio 8800 System

The following table lists the outbound IP ports currently used by the Polycom UC Software running on the RealPresence Trio 8800 system.

Outbound IP Port Connections to RealPresence Trio Systems

Inbound Port	*Type*	*Protocol*	*Function*	*Default*	*Configurable Port Number*
21	static	TCP	FTP Provisioning, Logs	On	No
22	static	TCP	SSH	On	No
53	static	UDP	DNS	On	No
67	static	UDP	DHCP Server	On	No
68	static	UDP	DHCP Client		No
69	static	UDP	TFTP Provisioning, Logs		No
80	static	TCP	HTTP Provisioning, Logs, Web Interface		No
123	static	UDP	NTP time server		No
389	static	TCP/UDP	LDAP directory query		No

Outbound IP Port Connections to RealPresence Trio Systems

Inbound Port	*Type*	*Protocol*	*Function*	*Default*	*Configurable Port Number*
443	static	TCP	HTTPS Provisioning, Logs, Web Interface		No
514	static	UDP	SYSLOG		No
636	static	TCP/UDP	LDAP directory query		No
2222	Dynamic (2222 - 2269)	TCP/UDP	RTP media packets	On	Yes, tcpIpApp.port.rtp.mediaPortRangeStart
2223	Dynamic (2222 - 2269)	TCP/UDP	RTCP media packets statistics	On	Yes, tcpIpApp.port.rtp.mediaPortRangeStart
5060		TCP/UDP	SIP signaling	On	
5061		TCP	SIP over TLS signaling	On	
5222	static	TCP	RealPresence Resource Manager: XMPP	Off	No
8001	static	TCP	HTTPS for modular room provisioning	On	Yes mr.deviceMgmt.port

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.