Technical Bulletin 60519

Simplified Configuration Improvements in Polycom® UC Software 3.3.0

This technical bulletin provides information about the configuration improvements introduced in Polycom® UC Software 3.3.0. It provides detailed information about the improvements so you can easily adopt them when you upgrade to the new release, without affecting the phone's capabilities.

This information applies to Polycom® SoundPoint® IP, SoundStation® IP, and VVX® 1500 phones running Polycom UC Software 3.3.0 or later.

The topics include:

Several changes associated with the improvements are not backward compatible with configuration files from earlier releases. Therefore, it is important to read this technical bulletin before you perform any large scale upgrades.

Introduction

Polycom UC Software 3.3.0 contains significant improvements to facilitate and simplify phone configuration and deployment. These improvements include:

- New functionality
- Improvements to existing functionality
- Improved debug capability
- Removal of complex features

The improvements to the configuration system are:

Safe Defaults for All Parameters

Benefit: It is no longer necessary to load the default **sip.cfg** and **phone1.cfg** files to ensure proper operation. This saves approximately 200 KB from being downloaded from the provisioning server for every phone, and will result in improved boot speed, particularly in 'power failure recovery' scenarios. Now, you only have to manage non-default parameters.

Implication: To take advantage of this change, you need to remove the **sip.cfg** and **phone1.cfg** templates from the **000000000000.cfg** or **<*MAC Address*>.cfg** files. For administrators that have created their own custom configuration files (for example, by combining **sip.cfg** and **phone1.cfg**), all parameters that are set to default values need to be removed.

Change is mandatory: Since the **sip.cfg** and **phone1.cfg** files are no longer distributed with the release, this change will need to be made.

Backward Compatibility: The change is not backward compatible. If you attempt to use pre-UC Software 3.3.0 configuration files with UC Software 3.3.0, some features may not work correctly. If your deployment contains legacy phones that do not run UC Software 3.3.0, you need to maintain the relevant **phone1.cfg** and **sip.cfg** configuration files for those phones. For detailed information on how to maintain these files, see Technical Bulletin 35311, *Supporting SoundPoint IP 300/301/500/501/600/601 and SoundStation IP 4000 phones with SIP 2.2.0 or SIP 3.2.0 and later releases*, available from http://www.polycom.com/support/voice/soundpoint_ip/VoIP_Technical_Bulletins_pub.html.

Many Parameter Changes Will Not Require a Re-boot to Take Effect

Benefit: When you change parameters using the phone's Web interface, the phone's user interface, or by a re-configuration invoked by a SIP-NOTIFY (Check-Sync), many configuration parameters will take effect immediately without requiring a re-boot. This allows for a much quicker implementation of configuration changes, and results in an improved experience for phone administrators that are using the phone for the first time or testing new features.

Implication: No administrator interaction is needed to take advantage of this feature.

Change is automatic: Parameters that do not need a re-boot will not cause a re-boot. This may potentially be an issue for people that depend on a parameter change to force a re-boot.

Changes to the following commonly used configuration parameters will no longer initiate a phone re-boot:

This list is not exhaustive and is intended for guideline purposes only. For more information, see the *Administrator's Guide for the Polycom® UC Software*, available from http://www.polycom.com/support/voice/index.html.

- SIP registration credentials—address, auth username, auth password
- Outbound proxy address
- Line display label
- Soft key remapping
- Static BLF configurations
- Applications API settings:
  - Browser
  - telNotification settings
- Logging levels
- Custom display backgrounds
- Custom idle display

Web Configuration Overrides File

Historically, changes made using the phone's Web interface or the phone's user interface have been stored in a common overrides file (**<*MAC Address*>-phone.cfg**). In UC Software 3.3.0, changes made using the phone's Web interface will be stored in a new file (**<*MAC Address*>-web.cfg**).

The phone will save an internal copy of this file and write a copy to the appropriate provisioning server. If the provisioning server is not available and a backup is not written, the internal version will be used on subsequent re-boots. However, if the Flash File System is cleared (for example, during a BootROM upgrade) these settings will be lost. Also, if the provisioning server is temporarily unavailable when a particular change is made, then that change will not be preserved for the next re-boot (since the server overrides file, if available, has higher precedence than the local version saved on the phone).

When you upgrade to UC Software 3.3.0, the settings previously stored in the overrides file will be retained in the overrides (and be treated as if they had been made from the phone's user interface). This may be an issue, since if the phone's Web interface is used to change one of these parameters after the upgrade, it will have lower precedence than the setting made before the upgrade. You can address this by deleting the setting in the **<*MAC Address*>-phone.cfg** overrides file.

If the phone is subsequently downgraded to a release earlier than UC Software 3.3.0, the settings changed using the phone's Web interface will be lost (since the phone will only know how to read the **<*MAC Address*>-phone.cfg** file). If

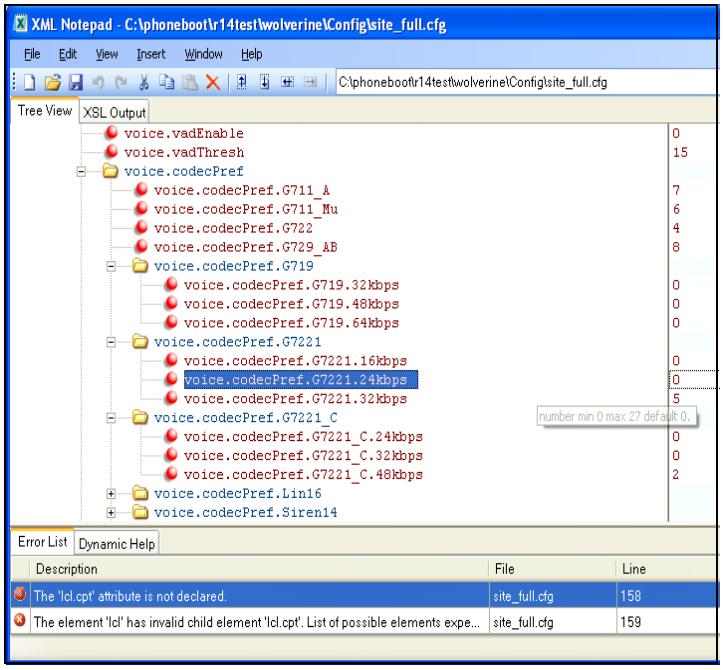
downgrade is necessary, you can combine the **<*MAC Address*>-phone.cfg** and **<*MAC Address*>-web.cfg** override files to preserve any settings made through the phone's Web interface while the phone was running UC Software 3.3.0.

To clear all the override settings, delete the **<*MAC Address*>-phone.cfg** and **<*MAC Address*>-web.cfg** files on the server, and clear the local configuration using the phone's user interface (select **Menu** > **Settings** > **Advanced** > **Password** > **Administration Settings** > **Reset to Defaults** > **Reset Local Configuration**, **Reset Web Configuration**, and (optionally) **Reset Device Settings**).

Broader Set of Configuration File Templates

Configuration file templates provided with this release have changed. Instead of **sip.cfg** and **phone1.cfg** files, the templates are split into a broader set of configuration files to accommodate various deployment scenarios. For more information about these files, see "Configuration File Templates" on page 11.

An XML schema definition for the configuration templates will be supplied. This file (**polycomConfig.xsd**), when used in conjunction with an XML editor, will detect errors and provide hints regarding appropriate parameter values (as shown in the figure below).



Improved Debugging Capability

Several features have been added to facilitate the detection and debugging of configuration file issues:

- Improved logging of configuration file issues.
- Configuration File Statistics menu.
- Warning message that displays when the phone boots if a significant number of errors are detected.
- Upload Configuration Administrator Menu option.

For more information on debugging, see "Configuration File Debug Features" on page 13.

Configuration Parameter Improvements

Many configuration parameters that were difficult to use have been improved. For more information about these improvements, see "Detailed Description of Configuration Parameter Changes" on page 18.

Many parameters are not backward compatible and will require configuration changes if they are used with pre-UC Software 3.3.0 releases. For more information, see "Detailed Description of Configuration Parameter Changes" on page 18.

New DHCP Option 60 Setting

The phone uses DHCP Option 60 to report information such as the manufacturer type and model number to the DHCP server. This information may be used to define the DHCP settings sent to the phone (for example, for a phone, DHCP Option 160 could be used to indicate the provisioning server address).

The format used by Polycom phones prior to UC Software 3.3.0 is incompatible with certain commonly used DHCP servers. New configuration settings can be used so the phone can deliver the DHCP Option 60 in a format that is compatible with the DHCP servers mentioned above. Use of this feature is described in Technical Bulletin 54041, *Using DHCP Vendor Identifying Options with Polycom® SoundPoint® IP, SoundStation® IP, and VVX® Phones*, available from http://www.polycom.com/support/voice/soundpoint_ip/VoIP_Technical_Bulletins_pub.html

The 'old' option will continue to be the factory default, since certain customers and partners already use this option. However, the default option will change after existing partners adapt their solutions to accommodate the new method.

Removal of Certain Features

Certain features, which were complex to use and support, have been removed, such as:

- Animated idle display

  This feature that allowed a dynamic logo to appear on the idle display has been removed. However, you can configure a static picture (for example, a logo) to appear on the idle display.

- Custom line and display icons

  The feature that allowed you to customize icons (for example, the icons that indicate call progress state) has been removed.

Phone Configuration Overview

This section provides an overview of how the phone configuration system operates when Polycom UC Software is deployed. The topics include:

Terminology

Term	Definition
Provisioning server	Server used to store the phone software, configuration files, custom image, and audio files and to save directory information and the backup of phone interface, web configuration, and phone log files. TFTP, FTP, HTTP and HTTPS access protocols are supported. **Note**: the various functions of the provisioning server can be separated into different physical or virtual (file structure) locations.
Overrides	User settings made through the phone's user interface or the phone's Web interface. A copy of the overrides is saved locally on the phone and on the provisioning server.
Phone user interface	Accessing the phone using the phone keypad, hard keys, and LCD display. Changes made to user accessible parameters and to administrative configuration are saved in **OVERRIDES/<*MAC Address*>-phone.cfg**
Phone Web interface	Accessing the phone over the Web. Browse to the phone's Web interface by entering http://<*Phone IP Address*>. Changes made to the phone configuration through the phone's Web interface are saved in **OVERRIDES/<*MAC Address*>-web.cfg**.
Configuration files	One or more files used to set configuration parameters for the phone. These are accessed from the provisioning server.

Term	Definition
Device Settings	Configuration settings made on the phone that are permanently saved in the phone's Flash memory. This includes most of the configuration parameters related to network address settings, provisioning server discovery, etc. (settings required for the phone to be operational in the IP network). The factory configures a default 'device settings.' Occasionally, these settings may change, and any changes are documented in technical bulletins. You can restore the default settings from the phone's user interface (select **Menu** > **Settings** > **Advanced** > **Administration Settings** > **Reset to Defaults** > **Reset Device Settings**).
Local Configuration Settings	Settings made using the phone's user interface to 'non device' settings (for example, ring tone selection, background selection, ringer volume, etc.). You can erase these parameters using the phone's user interface (select **Menu** > **Settings** > **Advanced** > **Administration Settings** > **Reset to Defaults** > **Reset Local Configuration**, or select **Menu** > **Settings** > **Advanced** > **Administration Settings** > **Reset to Defaults** > **Format File System**, and then delete the overrides file, ***<MAC Address>*-phone.cfg**, from the provisioning server). The Reset Local Configuration option will clear the overrides file on the provisioning server, provided that the server is available with write access when the action is carried out. **Note**: Re-loading the BootROM software performs a Format File System and will erase the local configuration settings if the overrides file has not been written.

Term	Definition
Web Configuration Settings	Settings made using the phone's Web interface to 'non device' settings (for example, ring tone selection, background selection, ringer volume, etc.). You can erase these parameters using the phone's user interface (select **Menu** > **Settings** > **Advanced** > **Administration Settings** > **Reset to Defaults** > **Reset Web Configuration**, or select **Menu** > **Settings** > **Advanced** > **Administration Settings** > **Reset to Defaults** > **Format File System**, and then delete the overrides file, ***<MAC Address>*-web.cfg**, from the provisioning server). The Reset Local Configuration option will clear the overrides file on the provisioning server, provided that the server is available with write access when the action is carried out. **Note**: Re-loading the BootROM software performs a Format File System and will erase the local configuration settings if the overrides file has not been written.

Configuration Basics

You can configure Polycom phones running UC Software using one or more of the following methods:

- Configuration Files
- Phone's Web interface
- Phone's user interface

If the same parameters are set using more than one method, the order of precedence is as follows:

1. Phone user interface (read from server)
2. Phone user interface (internal)
3. Web interface (read from server)
4. Web interface (internal)
5. Configuration files (read from server)
6. Configuration files (internal)

Internal files will be used when the server versions are not found during a boot. This can be either because the provisioning server is not available during boot, or if the override files were not written to the provisioning server (for example, because the server did not have write access, or the phone was pre-staged on a different server than it is currently accessing).

If you need to remotely manage devices, administrators are strongly encouraged to exclusively use configuration files to configure parameters. In these situations, it is recommended that you disable the phone's Web interface and change the administrator password for the phone's user interface to a value not known by users. All the phone configuration parameters may be configured using configuration files. The phone implements a configuration system that allows for more than one configuration file to be read and uses phone model/part number substitutions to facilitate the management of multiple types of phones and customer sites using a single provisioning solution. This is ideally suited to service provider deployment models.

In situations where only a few devices are being configured and managed, the phone's Web interface and/or the phone's user interface may be easier to use than setting up a configuration server. Not all of the settings on the phone can be configured from the phone's Web interface and phone's user interface.

The phone will save the parameters set using the phone's Web interface or phone's user interface in a local file and will attempt to write this overrides file to a provisioning server. You can upload these settings to a provisioning server through the phone's user interface (select **Menu > Settings > Advanced > Administration Settings > Upload Configuration**). If you want, you can clear these settings.

To clear the parameters set through the phone's Web interface or phone's user interface:

1 Do one of the following:

- From the phone's user interface, select **Menu > Settings > Advanced > Administration Settings > Reset to Defaults > Reset Local Configuration/Reset Web Configuration** and/or **Reset Device Settings** and/or **Format File System**.

or

- Re-load the BootROM.

2 Delete (or null out) the overrides file on the provisioning server.

For information on the parameters that you can set through the phone's Web interface and phone's user interface, see the *Administrator's Guide for the Polycom® UC Software*, available from http://www.polycom.com/support/voice/index.html.

Factory Default Configuration

Phones ship from the factory with a consistent software revision and default settings. Occasionally, this configuration is changed (about once a year) and customers are informed by technical bulletin of the changes. Customers may receive phones with different software and configuration settings due to delays introduced by the supply chain.

Software Release Default Configuration

Each software release (starting with UC Software 3.3.0) will have built-in valid defaults for all configuration parameters. You no longer have to configure each phone with a **sip.cfg** and **phone1.cfg** file within a customer deployment. You will only need to manage non-default parameters.

Phone Behavior When the Server is Unreachable

The phone will use the last available configuration (from its local storage) if the server is unreachable. The phone will boot to the best of its ability and will not fail due to missing configuration files.

Configuration File Templates

Various configuration file templates are available with each software release. These templates will be distributed with the software as part of the product release package.

The templates do the following:

- Document the syntax for configuration parameters.
- Document the default settings for each parameter.

You can use the template as-is for setting parameters, by simply changing any parameter settings. Polycom recommends that you use an XML editor to edit the files. Alternatively, you can cut and paste parameters you want to change from the template into a configuration file of your choice. This method reduces the size of the configuration file(s) that the phone needs to download.

Starting with UC Software 3.3.0, an XML schema definition for the configuration templates will be supplied. When used in conjunction with an XML editor, this file will detect errors and provide hints regarding appropriate parameter values. The schema file will be distributed as part of the product release package.

The templates have been designed to make it easy to deploy phones in different deployment scenarios, such as:

'Simple SIP' Device Deployment

This type of deployment is typically a conference phone that is installed for basic operation against a standard SIP call server. Two templates are provided:

- **reg-basic.cfg**
- **sip-basic.cfg**

'Typical' Hosted Service Provider Deployment

Deployments that use more than one line will require settings from other templates. The settings you require will depend on the call server implementation and the specific feature set that is implemented. Templates you may require include:

- **reg-basic.cfg**
- **sip-basic.cfg**
- **reg-advanced.cfg**
- **sip-interop.cfg**
- **site.cfg**
- **features.cfg**
- **applications.cfg** (if implementing applications using the browser and XML Control/Status API)
- **region.cfg** (optional, often used for non-North American settings)
- **video.cfg** (if using the VVX 1500 for video calls)

You can use files as-is, or merge relevant pieces of each one together to suit your needs.

'Typical' IP-PBX Deployment

Same as above.

HDX Video Integration Deployment

For deployments in which the SoundStation IP 7000 is deployed as an audio and control end point connected to a Polycom HDX device, you only need to configure a limited number of parameters. These parameters are contained in **video-integration.cfg**.

Certain parameters are not intended to be set during normal use of the product, but are useful for troubleshooting field issues and applying workarounds for inter-operability issues. These parameters are contained in the **techsupport.cfg** template file, which is restricted for use by qualified technical support personnel only.

Configuration File Debug Features

This section describes the debug features available in Polycom UC Software 3.30. The topics include:

Logging of Errors and Warnings

The following configuration file issues will be logged in the phone's application log file.

When the phone processes the configuration files, it will ignore unknown and invalid parameters and use the built-in defaults for those parameters. By doing so, there will be no configuration file issues that will cause the phone to deliberately re-boot or freeze.

- Pre-UC Software 3.3.0 parameters

  If any of the following parameters are detected, a log message will be generated to warn you that you are attempting to use configuration files created with an earlier release.

This is not an exhaustive list of parameters, since searching for all parameters will add time to the boot-up process.

  - tone.chord.ringer.1.freq.1
  - se.pat.callProg.1.name
  - ind.anim.IP_500.1.frame.1.duration
  - ind.pattern.1.step.1.state
  - feature.2.name
  - feature.9.name

- Unknown parameters

  If the phone detects a parameter that is not recognized, a log file message will be generated that details the parameter and the configuration file in which it was changed. Typically, this occurs when a configuration parameter is obsolete or there is a spelling or case-sensitive error.

- Invalid parameter values

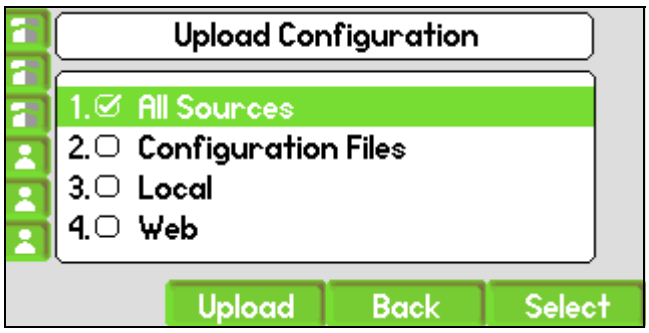
  If a parameter is detected with a value that is out of the valid range, a log file message will be generated.

The following example shows a log file depicting each of the warnings/errors.

```
0429202537|cfg  |*|00|Prm|0004f21acadb-user.cfg: Value "" for
configuration parameter roaming_buddies.reg is invalid, ignoring

0429202537|cfg  |*|00|Prm|0004f21acadb-user.cfg: Value "" for
configuration parameter roaming_privacy.reg is invalid, ignoring

0429202537|cfg  |4|00|Prm|ldap_ad_corp.cfg: Unknown parameter
"feature.19.name" found, ignoring.

0429202537|cfg  |4|00|Prm|ldap_ad_corp.cfg: Unknown parameter
"feature.19.enabled" found, ignoring.

0429202537|cfg  |4|00|Prm|sys.cfg: Unknown parameter "feature.1.name"
found, ignoring.

0429202537|cfg  |4|00|Prm|sys.cfg: Unknown parameter
"feature.1.enabled" found, ignoring.

0429202537|cfg  |4|00|Prm|sys.cfg: Unknown parameter "feature.2.name"
found, ignoring.

0429202537|cfg  |4|00|Prm|sys.cfg: Pre-SIP 3.3.0 parameter found,
please check your config.

0429202537|cfg  |4|00|Prm|sys.cfg: Unknown parameter
"feature.2.enabled" found, ignoring.

0429202537|cfg  |*|00|Prm|sys.cfg: Value "" for configuration parameter
log.level.change.so is invalid, ignoring

0429202537|cfg  |*|00|Prm|sys.cfg: Value "" for configuration parameter
mb.idleDisplay.refresh is invalid, ignoring

0429202537|cfg  |4|00|Prm|sys.cfg: Unknown parameter "mb.limits.nodes"
found, ignoring.
```

Ability to Upload Configuration Settings

From the phone's user interface, you can upload non-default settings to the provisioning server (root directory) by selecting **Menu > Settings > Advanced > Administration Settings > Upload Configuration**.



Files are uploaded as a flat file and include all parameters that are set using configuration files, the phone's user interface, and the phone's Web interface. Only the parameters that have non-default settings will be uploaded.

This feature is useful for:

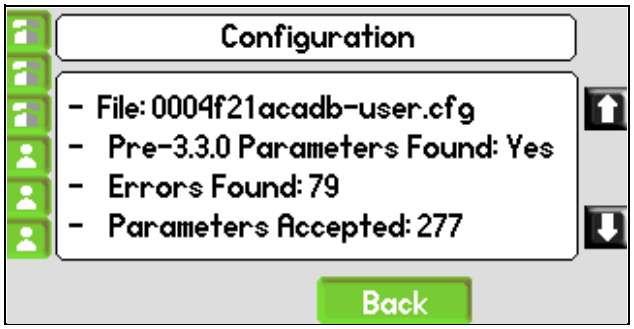
- Troubleshooting
- Cloning a phone configuration

  For example, you can configure a phone using the phone's Web interface, and then create configuration files for use on other phones.

Configuration File Statistics Menu

You can view statistics for the phone's configuration files from the phone's user interface (select **Menu > Status > Platform > Configuration**). The Configuration menu lists the configuration files being used and information such as number of parameters read and number of invalid parameters detected. This is useful information when you debug issues related to phone configuration.

The following figure shows a sample Configuration menu screen.



Warning Message at Boot-Up

If a large number of errors are detected during the configuration process, a warning message will display for five seconds immediately after the phone boots.

The following figure shows an example of a warning message.

If a configuration file contains any pre-UC Software 3.3.0 parameters, a warning message will display. For a list of pre-UC Software 3.3.0 parameters that will generate a log message, see "Logging of Errors and Warnings" on page 13.

Forward and Backward Compatibility

In general, configuration templates are not backward compatible to previous releases. However, steps have been taken to ensure an easy migration process for administrators that have deployed phones in the following deployment scenarios:

- Centralized provisioning, where:
  - Best practice recommendations from earlier releases have been followed.
  - **sip.cfg** and **phone1.cfg** files shipped with each release have been utilized.
  - Configuration files for all non-standard settings have been created.
- Centralized provisioning, where custom configuration files (often one file) have been created, and then edited with each release, to address any changes.
- Use of the phone's Web interface or phone's user interface as the primary method of configuring the device, where:
  - Factory defaults are used for all parameters not set by the user (that is, the phone has never contacted a provisioning server after it left the factory).
  - Phones are pre-staged to set certain parameters, and then the user sets parameters, such as SIP credentials, through the phone's Web interface or phone's user interface.

Polycom Configuration File Conversion Utility

Polycom is developing the Polycom Configuration File Conversion Utility (CFC utility) that can translate configuration files from pre-UC Software 3.3.0 to UC Software 3.3.0 format. The CFC utility will detect issues and changes and translate certain parameters. The CFC utility will upgrade (but not downgrade) configuration file changes. Before using the CFC utility, it is important to save a copy of your existing configuration. You can access the CFC utility from the Polycom Resource Center at http://portal.polycom.com/portal_web/polycom.portal?_nfpb=true&_pageLabel=service_support_product_support_voip&currentNavMenuItem=VoIP

Detailed Description of Configuration Parameter Changes

This section provides detailed information about the configuration of the following parameters:

For detailed information on all configuration parameters, see the *Administrator's Guide for the Polycom® UC Software*, available from http://www.polycom.com/support/voice/index.html.

Feature Parameters

Feature	Pre-UC Software 3.3.0 Naming (and default) in sip.cfg	UC Software 3.3.0 Naming in features.cfg (built-in default settings are illustrated)
Presence (using various WM formats)	feature.1.name="presence" feature.1.enabled="0"	feature.presence.enabled="0"
Instant Messaging (using WM signalling)	feature.2.name="messaging" feature.2.enabled="0"	feature.messaging.enabled="0"
Local Contact Directory	Feature.3.name="directory" Feature.3.enabled="1"	feature.directory.enabled="1"
Local Call Lists	Feature.4.name="calllist" Feature.4.enabled="1"	feature.callList.enabled="1"
Ring Download	Feature.5.name="ring-download" Feature.5.enabled="1"	feature.ringDownload.enabled="1"
Local Received Call List	Feature.6.name="calllist-received" Feature.6.enabled="1"	feature.callListRecieved.enabled="1"
Local Placed Call List	Feature.7.name="calllist-placed" Feature.7.enabled="1"	feature.callListPlaced.enabled="1"

Feature	Pre-UC Software 3.3.0 Naming (and default) in sip.cfg	UC Software 3.3.0 Naming in features.cfg (built-in default settings are illustrated)
Local Missed Call List	Feature.8.name="calllist-missed" Feature.8.enabled="1"	feature.callListMissed.enabled="1"
Allow URL dialing	Feature.9.name="url-dialing" Feature.9.enabled="1"	feature.urlDialing.enabled="1"
Call Park (using SIP-B signalling)	Feature.10.name="call-park" Feature.10.enabled="0"	feature.callPark.enabled="0"
Group Call Pickup (using SIP_B signalling)	Feature.11.name="group-call-pickup" Feature.11.enabled="0"	feature.groupCallPickup.enabled="0"
Directed Call Pickup (using SIP_B signalling)	Feature.12.name="directed-call-pickup" Feature.12.enabled="0"	feature.directedCallPickup.enabled="0"
Last Call Return (using SIP_B signalling)	Feature.13.name="last-call-return" Feature.13.enabled="0"	feature.lastCallReturn.enabled="0"
ACD Login/Out using either SIP-B or Feature Synchronization method	Feature.14.name="acd-login-logout" Feature.14.enabled="0"	feature.acdLoginLogout.enabled="0"
ACD Agent availability setting using either SIP-B or Feature Synchronization method	Feature.15.name="acd-agent-availability" Feature.15.enabled="0"	feature.acdAgentAvailability.enabled="0"
4-way conference and conference management UI (requires feature license). Varies by platform (see the latest *Release Notes*).	Feature.16.name="nway-conference" Feature.16.enabled="0"	feature.nWayConference.enabled="0"

Feature	Pre-UC Software 3.3.0 Naming (and default) in sip.cfg	UC Software 3.3.0 Naming in features.cfg (built-in default settings are illustrated)
USB Call Recording (requires feature license). Varies by platform (see the latest *Release Notes*).	Feature.17.name="call-recording" Feature.17.enabled="0"	feature.callRecording.enabled="0"
EFK and soft key configuration	Feature.18.name="enhanced-feature-keys" Feature.18.enabled="0"	feature.enhancedFeatureKeys.enabled="0"
Corporate Directory using LDAP (requires feature license). Varies by platform (see the latest *Release Notes*).	Feature.19.name="corporate-directory" Feature.19.enabled="0"	feature.corporateDirectory.enabled="0"

Ring Tone Parameters

UC Software 3.3.0 contains 24 built-in ring-tones.

Fourteen ring tones are configured using the phone's built-in synthesizer capability, and can be used as-is. It is possible, but not recommended, to modify them. Two ring styles (Low Trill Precedence and Ringback-style), are new in UC Software 3.3.0. They were not present in earlier releases.

The remaining 10 ring tones are intended for use by system administrators to add custom ring tones, typically utilizing an audio file (of type **.wav**).

Prior to UC Software 3.3.0, the phone supported 22 built-in ring tones and 10 administrator-defined ring tones. These ring tones were defined and referenced using an integer numbering scheme (from 1 to 22). In UC Software 3.3.0, integer-based referencing is replaced by string-based referencing. This is more intuitive and provides more flexibility for future releases when new built-in ring tones will be added.

The following table shows the ring tone mapping from pre-UC Software 3.3.0 to UC Software 3.3.0.

(Ringer Index)-X (Pre-UC Software 3.3.0)	Ringer XML tag STRING (UC Software 3.3.0)	Name/Order in Default Menu (UC Software 3.3.0)
1	ringer1	1-Silent Ring
2	ringer2	2-Low Trill
3	ringer3	3-Low Double Trill
4	ringer4	4-Medium Trill
5	ringer5	5-Medium Double Trill

(Ringer Index)-X (Pre-UC Software 3.3.0)	Ringer XML tag STRING (UC Software 3.3.0)	Name/Order in Default Menu (UC Software 3.3.0)
6	ringer6	6-High Trill
7	ringer7	7-High Double Trill
8	ringer8	8-Highest Trill
9	ringer9	9-Highest Double Trill
10	ringer10	10-Beeble
11	ringer11	11-Triplet
12	ringer12	12-Ringback-style
NA	ringer13	13-Low Trill Precedence
NA	ringer14	14-Ring Splash
13	ringer15	15-default (sampled1)
14	ringer16	16-default (sampled2)
15	ringer17	17-default (sampled3)
16	ringer18	18-default (sampled4)
17	ringer19	19-default (sampled5)
18	ringer20	20-default (sampled6)
19	ringer21	21-default (sampled7)
20	ringer22	22-default (sampled8)
21	ringer23	23-default (sampled9)
22	ringer24	24-default (sampled10)

For backward compatibilty reasons, the phone will recognize pre-UC Software 3.3.0 ringer index values as well as the new string values when used in the overrides file (for example, reg.X.ringType). This ensures that any user-selected ring tones are preserved during an upgrade to UC Software 3.3.0.

Sampled Ringers behave as follows:

- If nothing is configured, they display as "default" and play a ring tone that is unlike the other ring tones.
- If a **.wav** file is referenced, the ringer takes the name associated with the **.wav** file, or else displays Ringer<*number in list*> (for example, Ringer15). If the **.wav** file cannot be found or is not loaded, the name displays as Ringer<*number in list*>, but the default ring tone is played.
- By default, the configuration references sampled ringers as follows: sampled1=saf2, sampled2=saf3, etc.

Ring Tone Parameter Translations

This section describes the following ring tone parameters:

- Se.pat.ringer.XXX
- Tone.chord.ringer.XXX
- AlertInfo Type Ring Tones (VoIpProt.SIP.alertInfo.N.class and Se.rt.XXX)

Se.pat.ringer.XXX

Pre-UC Software 3.3.0 (sip.cfg) (values for X are shown in the table above)	UC Software 3.3.0 (site.cfg) (STRING values are shown in the table above)
se.pat.ringer.X.name="name"	se.pat.ringer.STRING="name"
se.pat.ringer.X.inst.y.type="chord|sampled|silence|branch"	Se.pat.ringer.STRING.inst.y.type="chord|sampled|silence|branch"
se.pat.ringer.X.inst.y.value=TONE-INDEX|SAMPLED-INDEX	se.pat.ringer.STRING.inst.y.value=TONE-NAME|SAMPLED-INDEX

Tone.chord.ringer.XXX

The tone.chord.ringer parameters have been changed so that a name, rather than an index, is used to reference parameters. No other changes have been made.

It is not expected that any customers will have changed the tone.chord parameters from the defaults.

Pre-UC Software 3.3.0 (sip.cfg)	UC Software 3.3.0 (site.cfg)
tone.chord.ringer.TONE-INDEX.freq.n	tone.chord.ringer.TONE-NAME.freq.n
tone.chord.ringer.TONE-INDEX.level.n	tone.chord.ringer.TONE-NAME.level.n
tone.chord.ringer.TONE-INDEX.onDur	tone.chord.ringer.TONE-NAME.onDur
tone.chord.ringer.TONE-INDEX.offDur	tone.chord.ringer.TONE-NAME.offDur
tone.chord.ringer.TONE-INDEX.repeat	tone.chord.ringer.TONE-NAME.repeat

The following table shows the mapping from TONE-INDEX (pre-UC Software 3.3.0) to TONE-NAME (UC Software 3.3.0).

TONE-INDEX (pre-UC Software 3.3.0)	TONE-NAME (UC Software 3.3.0)
1	F2
2	Gb2
3	G2
4	Ab2
5	A3
6	Bb3
7	B3
8	C3
9	Db3
10	D3
11	Eb3
12	E3
13	F3
14	Gb3
15	G3
16	Ab3
17	A4
18	Bb4
19	B4
20	C4
21	Db4
22	D4
23	Eb4
24	E4
25	F4
26 (spare_1single)	spare1
27 (spare_2single)	spare2
28 (spare_3single)	spare3

TONE-INDEX (pre-UC Software 3.3.0)	TONE-NAME (UC Software 3.3.0)
29 (spare_4single)	spare4
30 (spare_5single)	spare5
31	A3Major
32	Bb3Major
33	B3Major
34	C3Major
35	Db3Major
36	D3Major
37	Eb3Major
38	E3Major
39	F3Major
40	Gb3Major
41	G3Major
42	Ab3Major
43	A4Major
44 (spare_1multi)	spare6
45 (spare_2multi)	spare7
46	ringback
47 (spare_1pstn)	spare8
48 (spare_2pstn)	spare9
49 (spare_3pstn)	spare10
50 (spare_4pstn)	spare11
51	originalLow
52	originalHigh
53 (spare_1misc)	spare12
54 (spare_2misc)	spare13
55 (spare_3misc)	spare14
56	spare15
57	spare16
58	spare17

TONE-INDEX (pre-UC Software 3.3.0)	TONE-NAME (UC Software 3.3.0)
59	spare18
60	spare19
NA	splash

AlertInfo Type Ring Tones (VoIpProt.SIP.alertInfo.N.class and Se.rt.XXX)

These settings are used for SIP signalling (AlertInfo)-based ring tone selection. This is a commonly used feature, so you will most likely need to translate these parameters.

Identifying ring tone patterns using an index-referenced system has been replaced by a name-referenced system. The mapping of old INDEX_NUMBER to new NAME is detailed in the table below. Two new ringer definitions are added (precedence and splash).

The old parameter, voIpProt.SIP.alertInfo.N.class=INDEX_NUMBER, is replaced with voIpProt.SIP.alertInfo.N.class=NAME, where INDEX_NUMBER used to refer to se.rt.INDEX_NUMBER.AAA and now refers to se.rt.NAME.AAA

se.rt.N.name/type/... (N=) INDEX_NUMBER (pre-UC Software 3.3.0)	NAME (UC Software 3.30)
1	default
2	visual
3	autoAnswer
4	ringAutoAnswer
5	Internal
6	external
7	emergency
8	custom1
9	custom2
10	custom3
11	custom4
NA	precedence
	splash

For example:

- In pre-UC Software 3.3.0:

  voIpProt.SIP.alertInfo.1.value=http://127.0.0.1/Bellcore-dr2

  voIpProt.SIP.alertInfo.1.class=8

  se.rt.8.type=ring

  se.rt.8.ringer=5

- In UC Software 3.3.0:

  voIpProt.SIP.alertInfo.1.value=http://127.0.0.1/Bellcore-dr2

  voIpProt.SIP.alertInfo.1.class=custom1

  se.rt.custom1.type=ring

  se.rt.custom1.ringer=ringer5

Call Progress Tone Parameters

Call progress tone parameters use the same configuration methodology that ring tones use. As for ring tones, call progress tone parameters replace index-based referencing with name-based referencing.

This section describes the following call progress tone parameters:

- Se.pat.callProg
- Tone.chord.callProg.XXX

A sample excerpt from a **region.cfg** file showing call progress tone parameters is provided (see "Call Progress Tone Parameters Sample" on page 28).

Se.pat.callProg

Pre-UC Software 3.3.0 (sip.cfg) (values for X are displayed in the table below)	UC Software 3.3.0 (region.cfg) (STRING values are displayed in the table below)
se.pat.callProg.X.name="name"	se.pat.callProg.STRING="name"
se.pat.callProg.X.inst.y.type="chord|sampled|silence|branch"	Se.pat.callProg.STRING.inst.y.type="chord|sampled|silence|branch"
se.pat.callProg.X.inst.y.value=TONE-INDEX|SAMPLED-INDEX	se.pat.callProg.STRING.inst.y.value=TONE-NAME|SAMPLED-INDEX
Se.pat.callProg.X.inst.y.param (only used for X=12)	se.pat.callProg.STRING.inst.y.param=VALUE|NA

The following table shows the mapping from index (pre-UC Software 3.3.0) to STRING name (UC Software 3.3.0).

Call Progress Tone Index - X (pre-UC Software 3.3.0)	Ringer XML tag STRING (UC Software 3.3.0)	Name
1	dialTone	dial
2	busyTone	busy
3	ringback	ringback
4	reorder	reorder
5	stutter	stutter
6	callWaiting	call waiting
7	callWaitingLong	Long call waiting
8	confirmation	confirmation
9	howler	howler
10	recWarning	warning
11	msgWaiting	message waiting
12	alerting	alerting
13	intercom	intercom
14	bargeIn	bargein
15	secondaryDialTone	Secondary dial

Tone.chord.callProg.XXX

Call progress tones are built using standard tone definitions that are defined in the tone.chord parameters. In UC Software 3.3.0, call progress tone parameters replace index-based referencing with name-based referencing.

The following table shows the mapping from TONE-INDEX (pre-UC Software 3.3.0) to TONE-NAME (UC Software 3.3.0).

TONE-INDEX (X) (pre-UC Software 3.3.0)	TONE-NAME (UC Software 3.3.0)
1	dialTone
2	busyTone
3	ringback
4	reorder
5	stutter_3

TONE-INDEX (X) (pre-UC Software 3.3.0)	TONE-NAME (UC Software 3.3.0)
6	callWaiting
7	howler
8	recWarning
9	stutterLong
10	intercom

Make the above substitutions for each occurrence of the following parameters:

- tone.chord.callProg.X.freq.Y
- tone.chord.callProg.X.level.Y
- tone.chord.callProg.X.onDur
- tone.chord.callProg.X.offDur
- tone.chord.callProg.X.repeat

Call Progress Tone Parameters Sample

The call progress tones that are often configured on a per-country basis are Dial tone, Ringback tone, and Busy tone. These vary by country and have a very unique sound in each case.

The following is a sample of the tones for the UK. UK ringback uses a pattern that you will see in the se.pat.callProg.ringback parameter.

The following sample is extracted from **region.cfg**.

```
<se.pat.callProg.busyTone se.pat.callProg.busyTone.name="busy">
          <se.pat.callProg.busyTone.inst
se.pat.callProg.busyTone.inst.1.type="chord"
se.pat.callProg.busyTone.inst.1.value="busyTone"
se.pat.callProg.busyTone.inst.2.type=""
se.pat.callProg.busyTone.inst.2.value="">
          </se.pat.callProg.busyTone.inst>
        </se.pat.callProg.busyTone>
<se.pat.callProg.dialTone se.pat.callProg.dialTone.name="dial">
          <se.pat.callProg.dialTone.inst
se.pat.callProg.dialTone.inst.1.type="chord"
se.pat.callProg.dialTone.inst.1.value="dialTone"
se.pat.callProg.dialTone.inst.2.type=""
se.pat.callProg.dialTone.inst.2.value="">
          </se.pat.callProg.dialTone.inst>
        </se.pat.callProg.dialTone>


se.pat.callProg.ringback.name="ringback">
```

```
          <se.pat.callProg.ringback.inst
se.pat.callProg.ringback.inst.1.type="chord"
se.pat.callProg.ringback.inst.1.value="3"
se.pat.callProg.ringback.inst.2.type="silence"
se.pat.callProg.ringback.inst.2.value="1800"
se.pat.callProg.ringback.inst.3.type="chord"
se.pat.callProg.ringback.inst.4.type="branch"
se.pat.callProg.ringback.inst.3.value="3"
se.pat.callProg.ringback.inst.4.value="-3">
          </se.pat.callProg.ringback.inst>
        </se.pat.callProg.ringback>


<tone.chord.callProg.busyTone
tone.chord.callProg.busyTone.offDur="400"
tone.chord.callProg.busyTone.onDur="400"
tone.chord.callProg.busyTone.repeat="0">
          <tone.chord.callProg.busyTone.freq
tone.chord.callProg.busyTone.freq.1="400">
          </tone.chord.callProg.busyTone.freq>
          <tone.chord.callProg.busyTone.level
tone.chord.callProg.busyTone.level.1="-20"
tone.chord.callProg.busyTone.level.2="">
          </tone.chord.callProg.busyTone.level>
        </tone.chord.callProg.busyTone>


<tone.chord.callProg.dialTone tone.chord.callProg.dialTone.offDur="0"
tone.chord.callProg.dialTone.onDur="0"
tone.chord.callProg.dialTone.repeat="0">
          <tone.chord.callProg.dialTone.freq
tone.chord.callProg.dialTone.freq.1="350">
          </tone.chord.callProg.dialTone.freq>
          <tone.chord.callProg.dialTone.level
tone.chord.callProg.dialTone.level.1="-19"
tone.chord.callProg.dialTone.level.2="-19">
          </tone.chord.callProg.dialTone.level>
        </tone.chord.callProg.dialTone>
<tone.chord.callProg.ringback
tone.chord.callProg.ringback.offDur="200"
tone.chord.callProg.ringback.onDur="400"
tone.chord.callProg.ringback.repeat="2">
          <tone.chord.callProg.ringback.freq
tone.chord.callProg.ringback.freq.1="400"
tone.chord.callProg.ringback.freq.2="450">
          </tone.chord.callProg.ringback.freq>
          <tone.chord.callProg.ringback.level
tone.chord.callProg.ringback.level.1="-25"
tone.chord.callProg.ringback.level.2="-20">
          </tone.chord.callProg.ringback.level>
        </tone.chord.callProg.ringback>
```

Miscellaneous Tone Parameters

A similar translation is required to update miscellaneous tones. These include:

- instantMessage
- localHoldNotification
- messageWaiting
- negativeConfirm
- positiveConfirm
- remoteHoldNotification
- welcome

If any of these tones has been customized, you can use the **sip.cfg** (pre-UC Software 3.3.0) and **region.cfg** (UC Software 3.3.0) file templates to implement the translation for the se.pat.misc.XXX and tone.chord.misc.XXX parameters.

Platform Configuration Parameters

In prior releases, there were several configuration parameters that applied to several phone models. For example, parameters that included IP_330 also applied to the SoundPoint IP 320, 321, 330, 331, and 335 phone models.

In UC Software 3.3.0, all individual parameters can be applied to one phone model only by including the phone model string at the end of the parameter name. Where a generic parameter and a model-specific parameter exists, the model-specific parameter takes precedence. If there is no model-specific parameter, the global parameter will take precedence.

The specific parameters affected by this change are detailed in the following sections:

- "Platform Override Options" on page 31
- "Codec Configuration" on page 31
- "Hard Key Reassignment" on page 36
- "USB Recording Buffer Size" on page 37
- "Font Parameters" on page 38
- "Language List Parameters" on page 38
- "Voice.XXXX Parameters" on page 39

Platform Override Options

All mandatory platform-specific options existing in pre-UC Software 3.3.0 releases have been removed. However, you can override any configuration option for a particular platform by appending the platform label to the end of any configuration option.

For example, to disable the internal web server on all platforms except the SoundPoint IP 450, you would configure the parameters as follows:

httpd.enabled=”0”

httpd.enabled.SPIP450=”1”

The following table shows the platform label you can append to any configuration option.

Platform	Platform Label
SoundPoint IP 320	SPIP320
SoundPoint IP 321	SPIP321
SoundPoint IP 330	SPIP330
SoundPoint IP 331	SPIP331
SoundPoint IP 335	SPIP335
SoundPoint IP 450	SPIP450
SoundPoint IP 550	SPIP550
SoundPoint IP 560	SPIP560
SoundPoint IP 650	SPIP650
SoundPoint IP 670	SPIP670
SoundStation IP 5000	SSIP5000
SoundStation IP 6000	SSIP6000
SoundStation IP 7000	SSIP7000
VVX 1500	VVX1500

Codec Configuration

Prior to UC Software 3.3.0, you were required to configure codecs separately for each platform group. In UC Software 3.3.0, you can configure a simplified set of codec preferences for all platforms, improving consistency and reducing workload. If model-specific codec preferences are necessary, see “Platform Override Options” on page 31.

Pre-UC Software 3.3.0 Option	UC Software 3.3.0 Option
voice.codecPref.G711A	voice.codecPref.G711_A
voice.codecPref.IP_300.G711A	
voice.codecPref.IP_4000.G711A	
voice.codecPref.IP_6000.G711A	
voice.codecPref.IP_650.G711A	
voice.codecPref.IP_7000.G711A	
voice.codecPref.VVX_1500.G711A	
voice.codecPref.G711Mu	voice.codecPref.G711_Mu
voice.codecPref.IP_300.G711Mu	
voice.codecPref.IP_4000.G711Mu	
voice.codecPref.IP_6000.G711Mu	
voice.codecPref.IP_650.G711Mu	
voice.codecPref.IP_7000.G711Mu	
voice.codecPref.VVX_1500.G711Mu	
voice.codecPref.VVX_1500.G719.32kbps	voice.codecPref.G719.32kbps
voice.codecPref.VVX_1500.G719.48kbps	voice.codecPref.G719.48kbps
voice.codecPref.VVX_1500.G719.64kbps	voice.codecPref.G719.64kbps
voice.codecPref.IP_6000.G722	voice.codecPref.G722
voice.codecPref.IP_650.G722	
voice.codecPref.IP_7000.G722	
voice.codecPref.VVX_1500.G722	
voice.codecPref.IP_6000.G7221.16kbps	voice.codecPref.G7221.16kbps
voice.codecPref.IP_7000.G7221.16kbps	
voice.codecPref.VVX_1500.G7221.16kbps	
voice.codecPref.IP_6000.G7221.24kbps	voice.codecPref.G7221.24kbps
voice.codecPref.IP_7000.G7221.24kbps	
voice.codecPref.VVX_1500.G7221.24kbps	
voice.codecPref.IP_6000.G7221.32kbps	voice.codecPref.G7221.32kbps
voice.codecPref.IP_7000.G7221.32kbps	
voice.codecPref.VVX_1500.G7221.32kbps	

Pre-UC Software 3.3.0 Option	UC Software 3.3.0 Option
voice.codecPref.IP_6000.G7221C.24kbps	voice.codecPref.G7221_C.24kbps
voice.codecPref.IP_7000.G7221C.24kbps	
voice.codecPref.VVX_1500.G7221C.24kbps	
voice.codecPref.IP_6000.G7221C.32kbps	voice.codecPref.G7221_C.32kbps
voice.codecPref.IP_7000.G7221C.32kbps	
voice.codecPref.VVX_1500.G7221C.32kbps	
voice.codecPref.IP_6000.G7221C.48kbps	voice.codecPref.G7221_C.48kbps
voice.codecPref.IP_7000.G7221C.48kbps	
voice.codecPref.VVX_1500.G7221C.48kbps	
voice.codecPref.G729AB	voice.codecPref.G729_AB
voice.codecPref.IP_300.G729AB	
voice.codecPref.IP_4000.G729AB	
voice.codecPref.IP_6000.G729AB	
voice.codecPref.IP_650.G729AB	
voice.codecPref.IP_7000.G729AB	
voice.codecPref.VVX_1500.G729AB	
voice.codecPref.iLBC.13_33kbps	voice.codecPref.iLBC.13_33kbps
voice.codecPref.IP_6000.iLBC.13_33kbps	
voice.codecPref.IP_650.iLBC.13_33kbps	
voice.codecPref.IP_7000.iLBC.13_33kbps	
voice.codecPref.VVX_1500.iLBC.13_33kbps	
voice.codecPref.iLBC.15_2kbps	voice.codecPref.iLBC.15_2kbps
voice.codecPref.IP_6000.iLBC.15_2kbps	
voice.codecPref.IP_650.iLBC.15_2kbps	
voice.codecPref.IP_7000.iLBC.15_2kbps	
voice.codecPref.VVX_1500.iLBC.15_2kbps	
voice.codecPref.IP_7000.Lin16.16ksps	voice.codecPref.Lin16.16ksps
voice.codecPref.VVX_1500.Lin16.16ksps	
voice.codecPref.IP_7000.Lin16.32ksps	voice.codecPref.Lin16.32ksps
voice.codecPref.VVX_1500.Lin16.32ksps	

Pre-UC Software 3.3.0 Option	UC Software 3.3.0 Option
voice.codecPref.VVX_1500.Lin16.44_1ksps	voice.codecPref.Lin16.44_1ksps
voice.codecPref.IP_7000.Lin16.48ksps	voice.codecPref.Lin16.48ksps
voice.codecPref.VVX_1500.Lin16.48ksps	
voice.codecPref.VVX_1500.Lin16.8ksps	voice.codecPref.Lin16.8ksps
voice.codecPref.IP_6000.Siren14.24kbps	voice.codecPref.Siren14.24kbps
voice.codecPref.IP_7000.Siren14.24kbps	
voice.codecPref.VVX_1500.Siren14.24kbps	
voice.codecPref.IP_6000.Siren14.32kbps	voice.codecPref.Siren14.32kbps
voice.codecPref.IP_7000.Siren14.32kbps	
voice.codecPref.VVX_1500.Siren14.32kbps	
voice.codecPref.IP_6000.Siren14.48kbps	voice.codecPref.Siren14.48kbps
voice.codecPref.IP_7000.Siren14.48kbps	
voice.codecPref.VVX_1500.Siren14.48kbps	
voice.codecPref.IP_7000.Siren22.32kbps	voice.codecPref.Siren22.32kbps
voice.codecPref.IP_7000.Siren22.48kbps	voice.codecPref.Siren22.48kbps
voice.codecPref.IP_7000.Siren22.64kbps	voice.codecPref.Siren22.64kbps

If you configure a codec that a particular platform does not support, the device will ignore that preference and continue to the next configured preference. For example, using the default values, the highest-priority codec on a SoundPoint IP 650 will be G.722, since that model does not support Siren22, G.722.1C, or Siren14.

The table below illustrates how the default codec settings will be applied to each phone type. A default priority of 0 indicates that the codec will not be used. A default priority of 1 is highest priority. The “X” indicates that the codec is supported on the phone or phones that display in the particular column. A blank space indicates that codec is not supported on the phone or phones that display in the particular column.

Codec	Default Priority	VVX1500	SSIP7000	SSIP6000	SPIP335 SPIP450 SPIP550 SPIP560 SPIP650 SPIP670 SSIP5000	SPIP320 SPIP321 SPIP330 SPIP331
Siren22.64kbps	1		X			
Siren22.48kbps	0		X			
Siren22.32kbps	0		X			
G7221C.48kbps	2	X	X	X		
G7221C.32kbps	0	X	X	X		
G7221C.24kbps	0	X	X	X		
Siren14.48kbps	3	X	X	X		
Siren14.32kbps	0	X	X	X		
Siren14.24kbps	0	X	X	X		
G722	4	X	X	X	X	
G7221.32kbps	5	X	X	X		
G7221.24kbps	0	X	X	X		
G7221.16kbps	0	X	X	X		
G719.64kbps	0	X				
G719.48kbps	0	X				
G719.32kbps	0	X				
iLBC.13_33kbps	0	X	X	X	X	X
iLBC.15_2kbps	0	X	X	X	X	X
G711Mu	6	X	X	X	X	X
G711A	7	X	X	X	X	X
G729AB	8	X	X	X	X	X
Lin16.48ksps	0	X	X			
Lin16.44_1ksps	0	X				
Lin16.32ksps	0	X	X			
Lin16.16ksps	0	X	X	X	X	X
Lin16.8ksps	0	X				

Hard Key Reassignment

In UC Software 3.3.0, old platform group parameters are split into individual platforms. For the key reassignment feature, the old parameter names are incompatible with the new ones.

The following table shows the new individual platform names that replace the old platform group names. Key assignment names are unchanged. The only change is in platform names.

Platform Group Name (Pre-UC Software 3.3.0)	Individual Platform Name (UC Software 3.3.0)
key.IP_330.x.function	key.x.function.SPIP320 key.x.function.SPIP330 key.x.function.SPIP321 key.x.function.SPIP331 key.x.function.SPIP335
key.IP_430.x.function	N/A (platform no longer supported)
key.IP_450.x.function	key.x.function.SPIP450
key.IP_550.x.function	key.x.function.SPIP550 key.x.function.SPIP560
key.IP_650.x.function	key.x.function.SPIP650 key.x.function.SPIP670
key.IP_6000.x.function	key.x.function.SSIP6000
key.IP_7000.x.function	key.x.function.SSIP7000
New platforms	key.x.function.SSIP5000 key.x.function.VVX1500

For example, in a scenario where a site uses all 3xx phone models, and each phone has reassigned keys, the following parameters must be configured:

Pre-UC Software 3.3.0 Parameters	UC Software 3.3.0 Parameters
key.IP_330.x.function.prim	key.x.function.prim.SPIP320 key.x.function.prim.SPIP330 key.x.function.prim.SPIP321 key.x.function.prim.SPIP331 key.x.function.prim.SPIP335
key.IP_330.x.subPoint.prim	key.x.subPoint.prim.SPIP320 key.x.subPoint.prim.SPIP330 key.x.subPoint.prim.SPIP321 key.x.subPoint.prim.SPIP331 key.x.subPoint.prim.SPIP335

USB Recording Buffer Size

For this feature, the old parameter names are incompatible with the new ones. The generic default value applies to the VVX 1500 phone (and all future phones that have a USB recording feature). SoundPoint IP 650 and 670 phones now have individual overrides and are no longer grouped under the generic IP_650 name.

Pre-UC Software 3.3.0 Naming	UC Software 3.3.0 Naming (in debug.cfg)
usb.IP_650.recordBufSize.1=16	usb.recordBufSize.1.SPIP650=16 usb.recordBufSize.1.SPIP670=16
usb.IP_650.recordBufSize.2=25	usb.recordBufSize.2.SPIP650=25 usb.recordBufSize.2.SPIP670=25
usb.IP_650.recordBufSize.3=40	usb.recordBufSize.3.SPIP650=40 usb.recordBufSize.3.SPIP670=40
usb.VVX_1500.recordBufSize.1=48	usb.recordBufSize.1=48
usb.VVX_1500.recordBufSize.2=48	usb.recordBufSize.2=48
usb.VVX_1500.recordBufSize.3=48	usb.recordBufSize.3=48

Font Parameters

The font parameter has had platform groupings removed and replaced with generic platform override capability. Values of the font parameters must still be in the same format (<fontName>_<fontHeightInPixels>_<fontRange>.<fontExtension>), but can now be applied on a platform-by-platform basis.

Pre-UC Software 3.3.0 Parameters	UC Software 3.3.0 Parameters (in region.cfg)
font.IP_330.1.name 102x33	font.1.name.SPIP320 font.1.name.SPIP330 font.1.name.SPIP321 font.1.name.SPIP331 font.1.name.SPIP335
font.IP_400.1.name 132x46	N/A Platform(s) no longer supported
font.IP_500.1.name 160x80	N/A Platform(s) no longer supported
font.IP_600.1.name 320x160	font.1.name.SPIP550 font.1.name.SPIP560 font.1.name.SPIP650 font.1.name.SPIP670
N/A	font.1.name.SPIP450 font.1.name.SSIP5000 font.1.name.SSIP6000 etc.

Language List Parameters

For this feature, the old parameter names are incompatible with the new names. Previously, the parameter lcl.ml.lang.x.list was divided into two groups: IP_500 and IP_600. The IP_500 group contained platforms that were incapable of displaying Asian font character sets. The IP_600 group contained all other platforms. Now, the generic default value applies to all phones with a platform override parameter specified for each platform that is incapable of displaying Asian characters.

Pre-UC Software Parameters	UC Software 3.3.0 Parameters
lcl.ml.lang.IP_500.list = 2,3,4,5,6,7,8,9,12,13,14,15,16, 17,18	lcl.ml.lang.list.SPIP320 = 2,3,4,5,6,7,8,9,12,13,14,15,16,17,18 lcl.ml.lang.list.SPIP330 = 2,3,4,5,6,7,8,9,12,13,14,15,16,17,18 lcl.ml.lang.list.SPIP321 = 2,3,4,5,6,7,8,9,12,13,14,15,16,17,18 lcl.ml.lang.list.SPIP331 = 2,3,4,5,6,7,8,9,12,13,14,15,16,17,18 lcl.ml.lang.list.SPIP335 = 2,3,4,5,6,7,8,9,12,13,14,15,16,17,18
lcl.ml.lang.IP_600.list = All	lcl.ml.lang.list = All

Voice.XXXX Parameters

Several parameters of the type voice.XXX have been deprecated. These parameters are no longer configurable since the internal defaults have been specifically set for optimal performance. You should remove parameters of the the following type from configuration files:

- voice.aec.XXX
- voice.aes.XXX
- voice.ns.XXXX
- voice.agc.XXXX
- voice.rxEq.XXXX
- voice.txEq.XXXX
- voice.vad.XXXX
- voice.handset.XXXX
- voice.headset.XXXX

Some of the above parameters are still supported for normal use and for purposes of technical support. They are documented in the *Administrator's Guide for the Polycom® UC Software*, available from http://www.polycom.com/support/voice/index.html, and included in the **site.cfg** and **tech-support.cfg** configuration file templates.

Audio Parameters

Old parameter names that acted generally upon codecs featuring several options for bitrates have been removed. You must now configure the payloadSize, jitterBufferMin, jitterBufferMax, and jitterBufferShrink parameters for each bitrate individually.

This affects the following parameters (where x is one of: G719, Lin16, G7221, G7221C, Siren14, Siren22, iLBC):

- voice.audioProfile.x.**payloadSize**
- voice.audioProfile.x.**jitterBufferMin**
- voice.audioProfile.x.**jitterBufferMax**
- voice.audioProfile.x.**jitterBufferShrink**

Codecs without varying bitrate options are not affected (G711µ-law, G711a-law, G729AB, G722).

Pre-UC Software 3.3.0 Parameter	UC Software 3.3.0 Expanded Parameters
voice.audioProfile.Lin16.payloadSize	voice.audioProfile.Lin16.8ksps.payloadSize voice.audioProfile.Lin16.16ksps.payloadSize voice.audioProfile.Lin16.32ksps.payloadSize voice.audioProfile.Lin16.44_1ksps.payloadSize voice.audioProfile.Lin16.48ksps.payloadSize
voice.audioProfile.Lin16.jitterBufferMin	voice.audioProfile.Lin16.8ksps.jitterBufferMin voice.audioProfile.Lin16.16ksps.jitterBufferMin voice.audioProfile.Lin16.32ksps.jitterBufferMin voice.audioProfile.Lin16.44_1ksps.jitterBufferMin voice.audioProfile.Lin16.48ksps.jitterBufferMin
voice.audioProfile.Lin16.jitterBufferMax	voice.audioProfile.Lin16.8ksps.jitterBufferMax voice.audioProfile.Lin16.16ksps.jitterBufferMax voice.audioProfile.Lin16.32ksps.jitterBufferMax voice.audioProfile.Lin16.44_1ksps.jitterBufferMax voice.audioProfile.Lin16.48ksps.jitterBufferMax
voice.audioProfile.Lin16.jitterBufferShrink	voice.audioProfile.Lin16.8ksps.jitterBufferShrink voice.audioProfile.Lin16.16ksps.jitterBufferShrink voice.audioProfile.Lin16.32ksps.jitterBufferShrink voice.audioProfile.Lin16.44_1ksps.jitterBufferShrink voice.audioProfile.Lin16.48ksps.jitterBufferShrink
voice.audioProfile.G7221.payloadSize	voice.audioProfile.G7221.24kbps.payloadSize voice.audioProfile.G7221.32kbps.payloadSize voice.audioProfile.G7221.48kbps.payloadSize
voice.audioProfile.G7221.jitterBufferMin	voice.audioProfile.G7221.24kbps.jitterBufferMin voice.audioProfile.G7221.32kbps.jitterBufferMin voice.audioProfile.G7221.48kbps.jitterBufferMin
voice.audioProfile.G7221.jitterBufferMax	voice.audioProfile.G7221.24kbps.jitterBufferMax voice.audioProfile.G7221.32kbps.jitterBufferMax voice.audioProfile.G7221.48kbps.jitterBufferMax

Pre-UC Software 3.3.0 Parameter	UC Software 3.3.0 Expanded Parameters
voice.audioProfile.G7221.jitterBufferShrink	voice.audioProfile.G7221.24kbps.jitterBufferShrink voice.audioProfile.G7221.32kbps.jitterBufferShrink voice.audioProfile.G7221.48kbps.jitterBufferShrink
voice.audioProfile.G7221C.payloadSize	voice.audioProfile.G7221C.24kbps.payloadSize voice.audioProfile.G7221C.32kbps.payloadSize voice.audioProfile.G7221C.48kbps.payloadSize
voice.audioProfile.G7221C.jitterBufferMin	voice.audioProfile.G7221C.24kbps.jitterBufferMin voice.audioProfile.G7221C.32kbps.jitterBufferMin voice.audioProfile.G7221C.48kbps.jitterBufferMin
voice.audioProfile.G7221C.jitterBufferMax	voice.audioProfile.G7221C.24kbps.jitterBufferMax voice.audioProfile.G7221C.32kbps.jitterBufferMax voice.audioProfile.G7221C.48kbps.jitterBufferMax
voice.audioProfile.G7221C.jitterBufferShrink	voice.audioProfile.G7221C.24kbps.jitterBufferShrink voice.audioProfile.G7221C.32kbps.jitterBufferShrink voice.audioProfile.G7221C.48kbps.jitterBufferShrink
voice.audioProfile.Siren14.payloadSize	voice.audioProfile.Siren14.24kbps.payloadSize voice.audioProfile.Siren14.32kbps.payloadSize voice.audioProfile.Siren14.48kbps.payloadSize
voice.audioProfile.Siren14.jitterBufferMin	voice.audioProfile.Siren14.24kbps.jitterBufferMin voice.audioProfile.Siren14.32kbps.jitterBufferMin voice.audioProfile.Siren14.48kbps.jitterBufferMin
voice.audioProfile.Siren14.jitterBufferMax	voice.audioProfile.Siren14.24kbps.jitterBufferMax voice.audioProfile.Siren14.32kbps.jitterBufferMax voice.audioProfile.Siren14.48kbps.jitterBufferMax
voice.audioProfile.Siren14.jitterBufferShrink	voice.audioProfile.Siren14.24kbps.jitterBufferShrink voice.audioProfile.Siren14.32kbps.jitterBufferShrink voice.audioProfile.Siren14.48kbps.jitterBufferShrink
voice.audioProfile.Siren22.payloadSize	voice.audioProfile.Siren22.32kbps.payloadSize voice.audioProfile.Siren22.48kbps.payloadSize voice.audioProfile.Siren22.64kbps.payloadSize
voice.audioProfile.Siren22.jitterBufferMin	voice.audioProfile.Siren22.32kbps.jitterBufferMin voice.audioProfile.Siren22.48kbps.jitterBufferMin voice.audioProfile.Siren22.64kbps.jitterBufferMin
voice.audioProfile.Siren22.jitterBufferMax	voice.audioProfile.Siren22.32kbps.jitterBufferMax voice.audioProfile.Siren22.48kbps.jitterBufferMax voice.audioProfile.Siren22.64kbps.jitterBufferMax

Pre-UC Software 3.3.0 Parameter	UC Software 3.3.0 Expanded Parameters
voice.audioProfile.Siren22.jitterBufferShrink	voice.audioProfile.Siren22.32kbps.jitterBufferShrink voice.audioProfile.Siren22.48kbps.jitterBufferShrink voice.audioProfile.Siren22.64kbps.jitterBufferShrink
voice.audioProfile.iLBC.payloadSize	voice.audioProfile.iLBC.13_33kbps.payloadSize voice.audioProfile.iLBC.15_2kbps.payloadSize
voice.audioProfile.iLBC.jitterBufferMin	voice.audioProfile.Siren14.13_33kbps.jitterBufferMin voice.audioProfile.Siren14.15_2kbps.jitterBufferMin
voice.audioProfile.iLBC.jitterBufferMax	voice.audioProfile.Siren14.13_33kbps.jitterBufferMax voice.audioProfile.Siren14.15_2kbps.jitterBufferMax
voice.audioProfile.iLBC.jitterBufferShrink	voice.audioProfile.Siren14.13_33kbps.jitterBufferShrink voice.audioProfile.Siren14.15_2kbps.jitterBufferShrink
voice.audioProfile.G719.payloadSize	voice.audioProfile.G719.32kbps.payloadSize voice.audioProfile.G719.48kbps.payloadSize voice.audioProfile.G719.64kbps.payloadSize
voice.audioProfile.G719.jitterBufferMin	voice.audioProfile.G719.32kbps.jitterBufferMin voice.audioProfile.G719.48kbps.jitterBufferMin voice.audioProfile.G719.64kbps.jitterBufferMin
voice.audioProfile.G719.jitterBufferMax	voice.audioProfile.G719.32kbps.jitterBufferMax voice.audioProfile.G719.48kbps.jitterBufferMax voice.audioProfile.G719.64kbps.jitterBufferMax
voice.audioProfile.G719.jitterBufferShrink	voice.audioProfile.G719.32kbps.jitterBufferShrink voice.audioProfile.G719.48kbps.jitterBufferShrink voice.audioProfile.G719.64kbps.jitterBufferShrink

Idle Display Parameters (Bitmaps)

This feature allows you to display an administrator-supplied image on the phone's idle display.

This feature is not backward compatible.

Feature	Naming (and default) in sip.cfg (pre-UC Software 3.3.0)	Naming in .cfg (UC Software 3.3.0)
Phone displays an image on the idle display (usually a company or provider logo)	ind.idleDisplay.enabled (0)	bitmap.idleDisplay.name or bitmap.idleDisplay.name.x Where x = SPIP320, SPIP330, SPIP321, SPIP331, SPIP335, SPIP450, SPIP550, SPIP560, SPIP650, SPIP670, SSIP5000, SSIP6000, SSIP7000 or VVX1500 For example: bitmap.idleDisplay.name.SPIP450= “polycom.bmp” would display polycom.bmp on the idle display of the SoundPoint IP 450 phone bitmap.idleDisplay.name=”polycom.bmp” would display polycom.bmp on the idle display of all platforms
Sets the display size of the idle display on SoundPoint IP 320, 330, 321, 331, and 335 phones	ind.idleDisplay.mode (null)	bitmap.idleDisplay.mode • If set to 1 or Null, the idle display size is 87 x 11 pixels. • If set to 2, the idle display size is 87 x 22 pixels. • If set to 3, the idle display size is 102 x 22 pixels.
Sets the bitmap names used per platform. Starting with UC Software 3.3.0, animations are no longer used.	bitmap.x.y.name Where x = IP_330, IP_400, IP_450, IP_600, IP_4000, or IP_7000 and y = the bitmap number.	No longer used. Replaced with: bitmap.idleDisplay.name or bitmap.idleDisplay.name.x Where x = SPIP320, SPIP330, SPIP321, SPIP331, SPIP335, SPIP450, SPIP550, SPIP560, SPIP650, SPIP670, SSIP5000, SSIP6000, SSIP7000, VVX1500
Used for multiple bitmaps to create an animation	ind.anim.x.y.frame.z.bitmap (only added as needed)	No longer used
Used for multiple bitmaps to create an animation	ind.anim.x.y.frame.z.duration (only added as needed)	No longer used
Resource quota for bitmap image downloads	res.quotas.2.name – (bitmap) res.quotas.2.value – (10)	res.quotas.bitmap Needs to be adjusted based on total bitmap(s) file size

Previous Parameters	New Platform Specific Parameters
ind.anim.IP_330.29.frame.1.bitmap ind.anim.IP_330.29.frame.1.duration	bitmap.idleDisplay.name.SPIP320 bitmap.idleDisplay.name.SPIP330 bitmap.idleDisplay.name.SPIP321 bitmap.idleDisplay.name.SPIP331 bitmap.idleDisplay.name.SPIP335
ind.anim.IP_450.30.frame.1.bitmap ind.anim.IP_450.30.frame.1.duration	bitmap.idleDisplay.name.SPIP450
ind.anim.IP_600.29.frame.1.bitmap ind.anim.IP_600.29.frame.1.duration	bitmap.idleDisplay.name.SPIP550 bitmap.idleDisplay.name.SPIP560 bitmap.idleDisplay.name.SPIP650 bitmap.idleDisplay.name.SPIP670
ind.anim.IP_4000.29.frame.1.bitmap ind.anim.IP_4000.29.frame.1.duration	bitmap.idleDisplay.name.SSIP6000 **Note**: The SoundStation IP 4000 is no longer supported. However, the previous setting for the SoundStation IP4000 covered the SoundStation IP 6000.
ind.anim.IP_7000.29.frame.1.bitmap ind.anim.IP_7000.29.frame.1.duration	bitmap.idleDisplay.name.SSIP7000
ind.anim.IP_4000.29.frame.1.bitmap ind.anim.IP_4000.29.frame.1.duration	bitmap.idleDisplay.name.SSIP5000

Memory Allocation Parameters

The following memory-related parameters are updated in UC Software 3.3.0.

Pre-UC Software 3.3.0 Parameters	UC Software 3.3.0 Parameters
dir.local.volatile.4meg	dir.local.volatile.SPIP320 dir.local.volatile.SPIP330
dir.local.nonVolatile.maxSize.4meg	dir.local.nonVolatile.maxSize.SPIP320 dir.local.nonVolatile.maxSize.SPIP330
dir.local.volatile.8meg	dir.local.volatile
dir.local.nonVolatile.maxSize.8meg	dir.local.nonVolatile.maxSize
prov.fileSystem.ffs0.4meg.minFreeSpace	prov.fileSystem.ffs0.minFreeSpace.SPIP320 prov.fileSystem.ffs0.minFreeSpace.SPIP330
prov.fileSystem.ffs0.8meg.minFreeSpace	prov.fileSystem.ffs0.minFreeSpace

Pre-UC Software 3.3.0 Parameters	UC Software 3.3.0 Parameters
res.quotas.1.name="tone" res.quotas.1.value	res.quotas.tone
res.quotas.2.name="bitmap" res.quotas.2.value	res.quotas.bitmap
res.quotas.3.name="font" res.quotas.3.value	res.quotas.font
res.quotas.4.name="xmlui" res.quotas.4.value	res.quotas.xmlui
res.quotas.5.name="background" res.quotas.5.value	res.quotas.background
ramdisk.bytesPerBlock ramdisk.blocksPerTrack ramdisk.nBlocks ramdisk.nBlocks.IP_650	All three parameters have been replaced with ramdisk.size. If a custom ramdisk size has been used, the parameter for ramdisk.size can be calculated by multiplying the bytesPerBlock parameter with the appropriate nBlocks parameter (ramdisk.nBlocks.IP_650 is used for the SoundPoint IP 650 and SoundPoint IP 670 models). For most situations, the defaults should be used. If you need to use non-default parameters, see the *Administrator's Guide for the Polycom® UC Software*, available from http://www.polycom.com/support/voice/index.html.

Custom Indicators (LED Patterns)

The indicator parameters, ind.pattern.XXX, have been modified. Instead of using an index for each function, the name of the function is directly referenced in the parameter name.

The following table shows the mapping from index (pre-UC Software 3.3.0) to name (UC Software 3.3.0).

Pre-UC Software 3.3.0 Index	UC Software 3.3.0 Name
1	off
2	on
3	Wink
4	flash
5	flutter
6	brokenFlutter

Pre-UC Software 3.3.0 Index	UC Software 3.3.0 Name
7	Active
8	Offering
9	flashSlow
10	lockedOut
11	scaHeld
12	Held
13	remoteBusyOffering
14	powerSaving
15 (LED_CUSTOM1)	NOT SUPPORTED
16 (LED_CUSTOM2)	NOT SUPPORTED
17 (LED_CUSTOM3)	NOT SUPPORTED
18 (LED_CUSTOM4)	NOT SUPPORTED
19 (LED_CUSTOM5)	NOT SUPPORTED
20 (LED_CUSTOM6)	NOT SUPPORTED
21 (LED_CUSTOM7)	NOT SUPPORTED
22 (LED_CUSTOM8)	NOT SUPPORTED

The following parameters have been removed:

- ind.class.XXXX parameters
- Ind.gi.XXXX parameter
- Several Ind.led.XXXX parameters

Trademark Information

© 2010, Polycom, Inc. All rights reserved.

POLYCOM®, the Polycom “Triangles” logo and the names and marks associated with Polycom’s 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.