

OpenScape DeskPhone CP SIP

Service Information - Trace Guide

Unify SaS UCC PH CP DEV CL

Version: 1.2

Date: 11.08.2020

Unify Software and Solutions GmbH & Co. KG reserves the right to make changes and improvements to the products and any of the features of the products described in this document without prior notice. The contents of this document are provided "as is". Except as required by applicable law, no warranties of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose, are made in relation to the accuracy, reliability or contents of this document. Unify Software and Solutions GmbH & Co. KG reserves the right to revise this document or withdraw it at any time without prior notice.

WARNING: THIS DOCUMENT (OR DATA) CONTAINS INFORMATION THAT IS PROPRIETARY INTELLECTUAL PROPERTY AND/OR TRADE SECRET OF UNIFY GmbH & Co. KG AND MAY ONLY BE VIEWED BY AUTHORIZED PERSONS. UNAUTHORIZED VIEWING OR DISCLOSURE IS STRICTLY PROHIBITED. No part of this material may be copied or reproduced, in whole or in part, in any form (including photocopying and/or storage in any medium by electronic means and whether or not transiently or incidentally to some other use of this document) without the written permission of Unify Software and Solutions GmbH & Co. KG.

Copyright © Unify Software and Solutions GmbH & Co. KG, 2020
Otto-Hahn-Ring 6, 81739 Munich, Germany
All rights reserved.

History of Change

Version	Date	Author	Changes:
1.0	07.10.2019	Stefan Siewert	Initial release
1.1	18.12.2019	Stefan Siewert	Release
1.2	05.08.2020	Stefan Siewert	Update of trace sections

Table of Contents

1. Reason for this How-to	4
1.1 What it is NOT.....	4
2. Ethernet Traces	5
3. Phone Traces	5
3.1 Important information about phones traces	5
3.2 Activating Phone Traces	6
General Trace Configuration	7
Trace components	8
Trace Files	12
Trace Levels	13
3.3 Core Dump.....	14
3.4 Remote Trace Functionality.....	15
3.5 Saving of Phone Traces and Core Dumps	16
3.6 Check the Trace data.....	16
4. Trace Configuration Example scenarios	16
4.1 Best default trace configuration [Trace Level = TRACE].....	16
4.2 Ring Central configuration issues	16
4.3 Calls visible in Conversation list, even though no call is active (ghost call) [Trace Level = DEBUG].....	17
4.4 Speech path delay [Trace Level = DEBUG]	17
4.5 DSS LED delay (key module or local device) [Trace Level = DEBUG].....	17
4.6 Wrong item visible in Conversation list [Trace Level = DEBUG]	17
4.7 Mobility (problems during login/logoff) [Trace Level = DEBUG].....	18
4.8 802.1x issues	18

1. Reason for this How-to

The purpose of this how-to is to provide service personnel as well as self-maintaining customers with a guide to quality error reporting. It describes a best practice for providing useful traces for initial error reporting to either GO Service Desk and/or further to GVS.

It includes descriptions on how to capture ethernet-based traces as well as on activation, saving and deactivation of internal phone traces. Based on a few examples, it shows how to select which trace points to enable in order to match the error description of the user.

With this how-to in hand, the requester will be equipped to define the traces that are relevant for their topic.

1.1 What it is NOT

- NOT a guide that can be used to provide a set of complete trace data for every possible problem scenario and therefore avoid further requests from Development or GVS for additional trace data if needed.
- NOT a complete guide to fully understand the trace output.
- **NOT a replacement for a detailed error description during error reporting.**

2. Ethernet Traces

The very first trace that is helpful for finding the root cause of a problem with SIP phones is an ethernet trace.

We recommend using the tool “Wireshark” for capturing ethernet traces, and we recommend doing this directly at the phone with a network tap.

Other possibilities are Mirror Port on the network switch.

Using the mirror port feature of the phones’ internal network switch should only be used for topics that are directly reproducible. We do not recommend long-term tracing with this method.

If it is possible or required to run a long-term ethernet capture, we recommend configuring the Wireshark application to use multiple files, with each file having a maximum size of 50 megabytes.

Those traces are saved with a timestamp so that if users report that a particular issue happened at a specific time, the specific file and at least one of the files before and after the error time can be sent for further analysis.

3. Phone Traces

3.1 Important information about phones traces

Phone traces are stored in the phone’s RAM and are only saved in the phone’s flash if a reboot, that is initiated from an internal phone service, occurs. This applies to crash scenarios as well.

If a problem is reported, do not perform a power cycle reboot of the phone by pulling the power cable or LAN cable (in the case of power over ethernet). This will cause the traces to be lost!

If the phone stops responding but is not rebooting on its own, please use the remote trace feature described in chapter [3.4](#).

3.2 Activating Phone Traces

Log-in to the web-based management of the phone as administrator and select the option shown below:

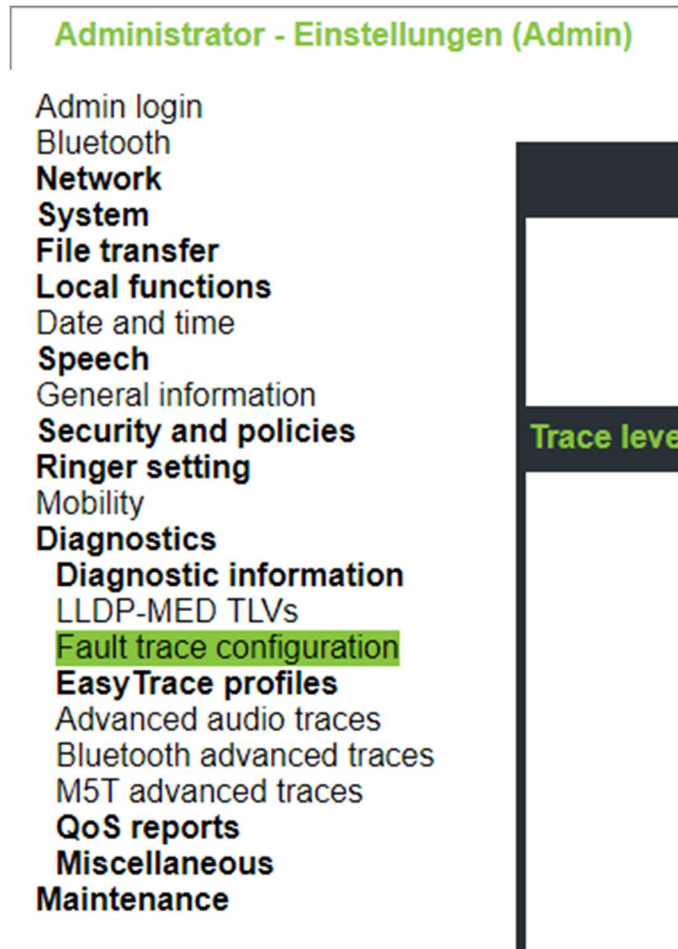
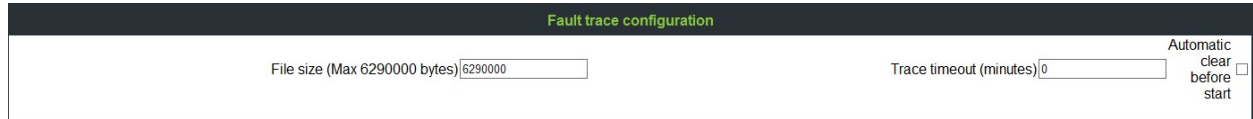


Figure 1: SIP WBM picture

The fault trace configuration page is divided into three parts: general configuration, trace components and trace files.

General Trace Configuration

The general configuration of the traces lets you configure the file size of the trace file, a timeout for the traces, and it will let you delete all previous data in the trace file.



Fault trace configuration	
File size (Max 6290000 bytes) <input type="text" value="6290000"/>	Trace timeout (minutes) <input type="text" value="0"/>
Automatic clear before start <input type="checkbox"/>	

The file size can range between 1 MB (1048576 bytes) and 6 MB (6290000 bytes). It is recommended to use a file size of 3 MB (3145728 bytes). Increasing the file size above this limit is not recommended for OpenScape Desk Phone CP200, OpenScape Desk Phone CP205 and OpenScape Desk Phone CP100 due to the memory consumption.

The trace timeout will clear all active traces after the configured timeout. A timeout value of 0 disables the timeout. Please only configure a trace timeout if you are sure that the error is captured within that timeframe.

On activation of traces the “Automatic clear before start” check box should be selected before collecting new trace data, as it will remove old logging data after the Submit button is clicked.

If, for example, the user reports that after 2 days of usage the phone shows an error and the error is persisting, then it might be useful to activate the traces and let the user perform the actions that show the error and save the traces after successful reproduction of the error.

However, in order to get the best traces, the traces must be activated before the error has happened. Otherwise, it will only show the consequences of the fault but not what has caused it.

The root cause analysis is not always possible with traces activated after the error has happened.

Trace components

Fault trace configuration

File size (Max 6290000 bytes)

Trace timeout (minutes)

Automatic clear before start

Trace levels for components

802.1x service	OFF	▼	Administration	OFF	▼
Application framework	OFF	▼	Application menu	OFF	▼
Broadsoft service	OFF	▼	Bluetooth service	OFF	▼
Call view	TRACE	▼	Certificate management	OFF	▼
Circuit service	OFF	▼	Clock service	OFF	▼
Communications	OFF	▼	Component registrar	OFF	▼
ConversationAPI	OFF	▼	CSTA service	TRACE	▼
Data access service	OFF	▼	Digit analysis service	OFF	▼
Directory service	OFF	▼	DLS client management	OFF	▼
Exchange service	OFF	▼	GPALAudio Core	OFF	▼
GPALAudio Framework	OFF	▼	Health service	OFF	▼
Instrumentation service	OFF	▼	Journal service	OFF	▼
Media control service	OFF	▼	Media recording service	OFF	▼
Mobility service	OFF	▼	OBEX service	OFF	▼
OpenScape UC service	DEBUG	▼	OpenStage client management	OFF	▼
Password management service	OFF	▼	Performance marks	OFF	▼
Physical interface service	OFF	▼	Security log service	OFF	▼
Service framework	OFF	▼	Service registry	OFF	▼
Sidcar service	OFF	▼	SIP call control	TRACE	▼
SIP messages	TRACE	▼	SIP signalling	TRACE	▼
SIP M5T stack	OFF	▼	Team service	OFF	▼
Tone generation service	OFF	▼	Transport service	OFF	▼
vCard parser service	OFF	▼	Video service engine	OFF	▼
Voice engine service	OFF	▼	Web server service	OFF	▼

The following trace components can be activated to various levels for the phone trace. Some traces might not be shown for various phone types.

802.1x service

This is for port security. 802.1X prevents unauthorized network access until appropriate credentials are supplied to access the network.

Administration

This deals with the changing and setting of configuration items in User and Admin menus, managing mobility notification popups, and call-forwarding activities.

Application framework

All applications within the phone (e.g. Call view, Call log or Phonebook) are run within the application framework. It is responsible for switching between different applications and bringing them into and out of focus as appropriate. It's also responsible for widget problems in GUI, such as incorrect display items.

Application menu (legacy trace setting)

This is where applications to be run on the phone can be started and stopped.

Bluetooth service

This handles the Bluetooth interactions between external Bluetooth devices and the phone.

Broadsoft service

Traces communication between phone and the XSI server, if configured.

Call view

This handles the representation of telephone calls on the phone screen.

Certificate management

This handles the certificates for WBM and SPE and 802.1x.

Clock service

This service traces all clock related messages, including SNTP and time taken from the OpenScape server.

Communications

This is for tracing all signaling between phone and system (keep alive, key presses and call related).

Component registrar

This handles data relating to the type of the phone, either OpenScape DeskPhone CP High (400/600) or OpenScape DeskPhone CP Low (200).

ConversationAPI

The important part of the Conversation List; it combines the call history information from Call Log and the contact information in the phonebook.

CSTA service

Any CSTA messages are handled by this service. CSTA messages are used within the phone by all services as a common call progression and control protocol.

Data access service

This service allows other services to access the data held within the phone database.

Digit analysis service

This analyzes and modifies digit streams that are sent and received by the phone, e.g. canonical conversion.

Directory service

This performs a look-up service for data in the phonebook, LDAP or System Phonebook, trying to match incoming and outgoing numbers with entries in the phonebook, LDAP or System Phonebook.

DLS client management

These traces display the messages between OpenStage Phone and DLS server and operations in DLS service.

Exchange service

Traces communication between phone and the exchange server, if configured.

GPALAudio Core

Core audio component traces with low-level information.

GPALAudio Framework

Advanced audio component traces with low-level information.

Health service

This monitors other parts of the phone for diagnostic purposes and provides a logging interface for the other services in the phone.

Instrumentation service

This is used by the Husim phone tester to exchange data with the phone for remote control, testing, and monitoring purposes.

Journal service

The journal service is responsible for saving and retrieving call history information which is used by the call log application.

Media control service

This service provides the control of media streams (voice, tones, ringing, etc.) within the phone.

Media recording service

This is a layer of software between the media control service and the tone generation and voice engine services. It is also involved in the switching of audio devices, such as the handset and loudspeaker.

Mobility service

This handles the mobility feature whereby users can log onto different phones and have them configured to their own profile.

OpenScape UC service

Traces communication and events between phone and the OpenScape UC server, if configured.

OpenStage client management (also relevant for CP devices)

This trace allows you to control the data flow of the system in case a configuration item on Local Menu, WEBM or DLS is deleted/updated/added. It resides in the middle of all services and provides interface to all other services for data management.

Password management service

This is used to verify passwords used in the phone.

Performance marks

This traces records when hook switch key is pressed. Physical button presses are also shown.

Physical interface service

This handles any interactions with the phone via the keypad, mode keys, fixed feature buttons, click wheel and slider. Also LED indicators are managed by this service.

Security log service

This handles the entries that are sent to the security log.

Service framework

This is the environment within which other phone services operate. It is involved in the starting and stopping of services.

Service registry

This keeps a record of all services that are currently running inside the phone.

Sidecar service

This handles interactions between the phone and any attached sidecars. In case of problems with key handling on sidecar please activate.

SIP call control

This contains the call model for the phone and is associated with telephony and call handling.

SIP messages

This traces the SIP messages that are exchanged by the phone.

SIP signaling

This is involved in the creation and parsing of SIP messages and communicates directly with the SIP stack.

SIP M5T stack

SIP stack internal trace messages.

Team service

This is primarily concerned with keyset operation.

Tone generation service

This service handles the generation of the tones and ringers on the phone. High-quality ringers or wav/MP3/MIDI files are not handled by this service.

Transport service

The transport service provides the IP (LAN & Wi-Fi) interface between the phone and the network.

Video service engine

This is for the video part of the phone.

Voice engine service

This provides a switching mechanism for voice streams within the phone. It is also involved in QDC, high-quality ringers/wav/MP3/MIDI and voice instrumentation.

Web server service

Traces the webserver of the web-based management of the device.



Figure 2: trace file links

Trace Files

Pictured above are the links to the various trace files on the phone. The information that every file includes is described below in the following terms.

Trace File

The trace file contains the latest trace information from the phone application. It is either started at the last reboot or if the configured file size was exceeded previously. See also “Old Trace File”.

Saved Trace File

The saved trace file contains the last 1MB of information that was present in the “trace file” before the last reboot that the phone performed and that wasn’t a power cycle or software upgrade. On a software upgrade this file is deleted as memory space is required for a software upgrade.

Upgrade Trace File

The upgrade trace file contains information about the latest software upgrade process.

Old Trace File

The old trace file only contains information if the trace file had exceeded the configured file size. In this case, the previous trace file is saved in phone RAM as old trace file and a new trace file is created. If an old trace file already existed, it will be deleted before the trace file is renamed. Information from the old trace file are never saved to flash even if the trace file hasn’t exceeded 1 MB before the last reboot.

Syslog File

The syslog file includes output from the underlying Linux and network stack.

Old Syslog File

The same as for old trace file, but it contains the syslog trace information.

Saved Syslog File

The same as saved trace file, but it contains the syslog trace information.

Database File

It contains the phone configuration in a SQL database.

Upgrade Error File

Information about errors during the last upgrade can be found here. Getting the information that several processes are not killed during upgrade is normal and not an actual error.

HPT Remote Service Log File

Logging of usage of the HPT remote service tool can be found here.

Exception File

The exception file contains a history of restarts that the phone has performed, including basic information about the reason for the reboot or crash. If the phone crashed, a basic stack trace output is shown for this restart.

Old Exception File

Not yet used.

Security Log File

Information about security information, such as wrong password attempts, expired passwords, and access attempts

Trace Levels

Various trace levels are available for each component.

For the purpose of this document the only two relevant trace levels are:

OFF

This will turn the trace component off.

DEBUG

This will activate the trace with all available logging.

The other trace levels should only be set on request (see trace examples), or if they are set by an easy trace profile.

After a problem has been traced and data has been collected using the DDC Tool [3.5](#), please set all traces to OFF. This can also be done by the Easy Trace Profile "Clear all Profiles".

3.3 Core Dump

The core dump is important for us to see what is going wrong. Normally the phone automatically generates a core dump if the phone crashes.

Core dumps are linked on this page:

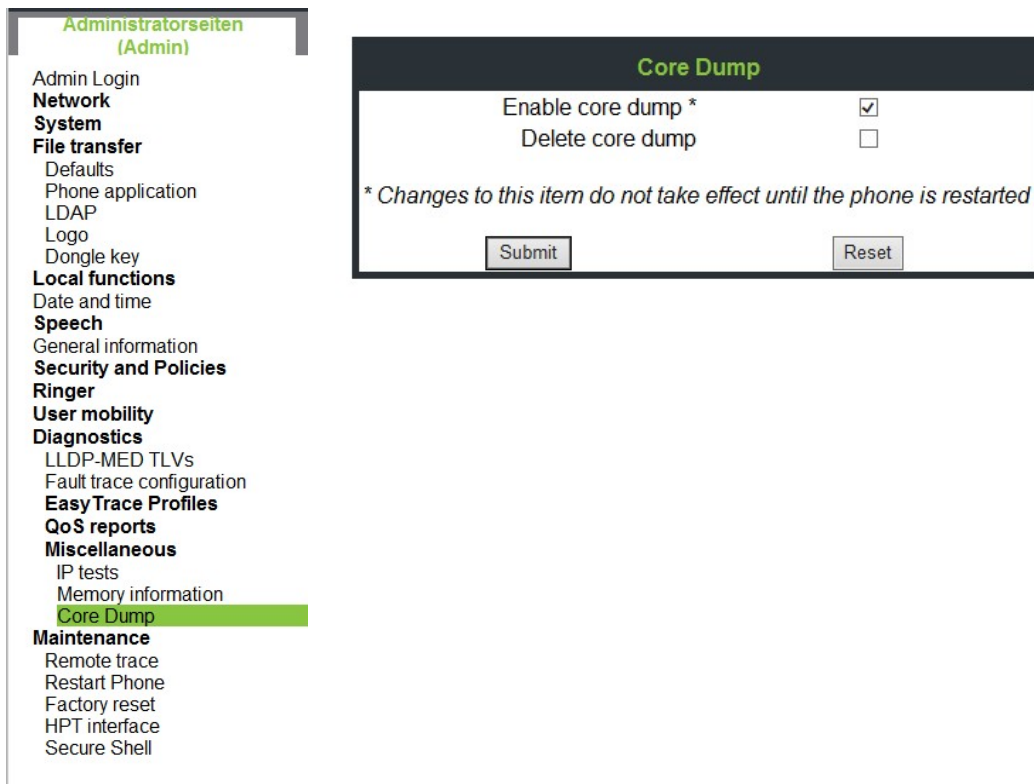


Figure 2: SIP WBM picture

On this page there are two parameters that could be set:

Enable core dump: This will enable that the memory dumps are written on a crash. **This parameter is enabled by default and should NOT be disabled!**

Delete core dump: This will delete all current core dumps that are available on the phone.

3.4 Remote Trace Functionality

With the remote trace the phone will send the trace output that is normally written into the trace file to the configured syslog server, and in addition it is written into the phone's memory as well.

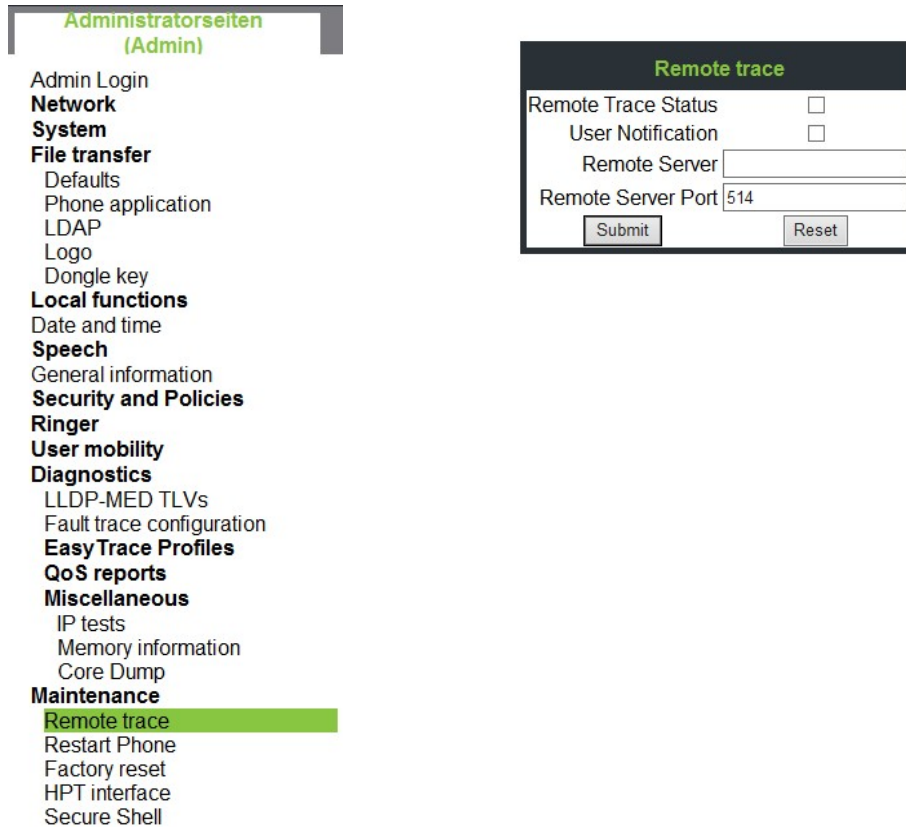


Figure 3: SIP WBM picture

Remote Trace Status

If this is set the remote trace is activated. If it is not set it is deactivated.

User notification

If the user notification is set the user gets informed about the ongoing tracing by an icon on the phone display (only shown on CP400/600).

Remote Server

This is the IP of the server where the syslog is running.

Remote Server Port

This is the UDP port at which the remote syslog server is configured to listen. This may be different based on customer environment.

Please note that the remote trace messages are NOT encrypted!

Recommended syslog server is Syslog-NG but others may work as well. A configuration file for the trace output on this server is available at Unify Expert Wiki:

http://wiki.unify.com/wiki/OpenStage_HFA_FAQ#Config_File_Syslog_NG

IMPORTANT: It was observed that the 3CDaemon Syslog Server is not saving all syslog Information the phone sends out. **DO NOT USE THE 3CDaemon Syslog Server.**

3.5 Saving of Phone Traces and Core Dumps

It is not required to download every single trace file by hand.

There is a service tool available on the software supply server or partner portal that will download all traces with a single step. This tool is called “Diagnostic Data Collector” or short: DDC-Tool.

With this tool you only need the IP address + Administrator password of the device to download all trace data and configuration information with a single click into a single .zip file.

Please note: with OpenScape Desk Phone CP HFA only V5 of the DDC Tool is supported. Don't use the V4 of DDC as it will miss important parts of traces.

Please always use the DDC-Tool to avoid missing required trace files and avoid questions about basic configuration settings. Even if using the remote trace functionality.

As previously noted, even the use of the DDC-Tool will not free you from providing a quality error description!

3.6 Check the Trace data

If using the DDC-Tool to collect traces, they are stored in a single .zip file. Before sending those data to GVS, please check them for following points as well:

1. The error time is present in the traces by checking the timestamp in the trace files including old* or saved*.
2. The requested traces are activated by checking the _device_configuration.html (which is part of the .zip file)

4. Trace Configuration Example scenarios

4.1 Best default trace configuration [Trace Level = TRACE]

- SIP Messages
- SIP signaling
- SIP call control
- Call View
- CSTA Service

4.2 Ring Central configuration issues [Trace Level = DEBUG]

- Certificate management
- DLS client management

4.3 Calls visible in Conversation list, even though no call is active (ghost call) **[Trace Level = DEBUG]**

- Call view
- ConversationsAPI
- Communications
- Journal service

4.4 Speech path delay **[Trace Level = DEBUG]**

- **Wireshark (LAN trace)**
- GPALAudio Core
- SIP messages
- Tone generation service
- Call view
- ConversationAPI
- Directory service
- GPALAudio Framework
- Media control service
- Performance marks
- Voice engine service
- SIP Signaling
- SIP Call Control
- Communications

4.5 DSS LED delay (key module or local device) **[Trace Level = DEBUG]**

- Call view
- Media control service
- Physical interface service
- Sidecar service
- SIP messages
- Tone generation service
- Team service
- SIP Signaling
- SIP Call Control
- Communications

4.6 Wrong item visible in Conversation list **[Trace Level = DEBUG]**

- Communications
- Call View
- ConversationAPI
- Data access service
- Directory service
- Journal Service
- OpenStage client management

4.7 Mobility (problems during log-in/log-off) [Trace Level = DEBUG]

- SIP Messages
- SIP signaling
- SIP call control
- Mobility service
- Administration
- DLS Client management

4.8 802.1x issues

- **Wireshark (LAN trace)**
- 802.1x service
- Transport service