Sangoma Technologies Inc.

NetBorder Call Analyzer (NCA)

User Guide

v2.0.0
August 27, 2010

Version: 2.0.0

Table of Contents

About this document

The NetBorder Call Analyzer is an open, software-based *VoIP (Voice over Internet Protocol)* product that provides enhanced *Call Progress Analysis (CPA)* services.

Audience

This document is intended for application developers and system administrators who manage and interface with the NetBorder Call Analyzer.

Prerequisites

This guide is intended for installers and advanced users. Prior knowledge of *IP (Internet Protocol)* networks is required.

This guide assumes:

- You have planned and/or managed the requirements of your VoIP and IP network.

- You have a working knowledge of Windows operating systems, the Internet, and graphical user interfaces.

For information on system requirements, see System requirements on page 18.

Organization

This document is organized as follows:

Section	*Title*	*Description*
Chapter 1	Product overview	Provides a description of the NetBorder Call Analyzer, as well as an explanation of the system architecture.
Chapter 2	Installation	Describes how to install (and uninstall) the NetBorder Call Analyzer software.
Chapter 3	Getting started	Gets you started using the NetBorder Call Analyzer.
Chapter 4	Call flow fundamentals	Describes the structure of SIP messages and provides sample call flows.
Chapter 5	Configuring the application	Describes how to configure the NetBorder Call Analyzer, including enabling logging.
Chapter 6	Troubleshooting	Provides solutions to key troubleshooting issues.
Appendix A	Glossary	Contains a list of terms, abbreviations and acronyms used in this guide.
Appendix B	Configuration parameters	Provides a comprehensive list of parameters, with a brief description, for the main configuration files.
Appendix C	Logging configuration	Contains general information about logging and logging configuration.
Appendix D	SIP response codes	Lists and describes the most frequent SIP response codes you will encounter.
Appendix E	Sample SIP messages	Provides sample call flows and SIP request and response messages.

Related documentation

Together with this guide, you may also want to reference the following additional Sangoma documentation:

- *NetBorder Call Analyzer Release Notes*: For a list of supported features, limitations, and known issues with the current release.

For the latest news and information on our products and on current as well as upcoming releases, visit the Sangoma Technologies Inc. website at www.sangoma.com.

Reference material

Commercial documentation on related technologies and applications is widely available from a number of sources. In addition, you may find the following specific information helpful.

SIP RFCs

In March 1999, *SIP (Session Initiation Protocol)* was defined in *RFC (Request for Comments)* 2543 by the *Multiparty Multimedia Session Control (MMUSIC)* Working group of the *Internet Engineering Task Force (IETF).* In June 2002, the IETF published a new SIP RFC (RFC 3261). The NetBorder Call Analyzer is fully compliant with RFC 3261.

You can find all RFCs online at http://www.ietf.org/rfc/rfc[xxxx].txt, where [xxxx] is the number of the RFC; for example, http://www.ietf.org/rfc/rfc3261.txt.

To search by topic, visit http://www.rfc-editor.org/rfcsearch.html.

Text and writing conventions

This document uses the following text and writing conventions:

- **Boldface** indicates menu items, or selections you make such as from a drop-down list or right-click context menu.

  For example: In the Services list, right-click “Sangoma NetBorder Call Analyzer”, and select **Properties** from the context menu.

- *Italics* indicate book titles, parameters and elements, file and path names, as well as terms introduced for the first time, which are usually spelled out and followed by their acronym or abbreviation in parentheses.

  For example: *VoIP (Voice over* Internet *Protocol)*

- `Courier New` indicates commands and keywords that you enter literally as shown, and on-screen output such as prompts and system messages.

  For example: `ipconfig /all`

- [Square brackets] indicate values that you replace, which are often followed by an explanation of what is required. Do not type the brackets when entering the command.

  For example: *[NETBORDER_INSTALLDIR]\config\cpa_media_uas.properties*

  where *[NETBORDER_INSTALLDIR]* is the root folder of the installation (for example, *C:\Program Files\Sangoma NetBorder Platform 2.0\config\cpa_media_uas.properties*).

In addition, note boxes, tips and cautions point out areas of special interest or concern. These boxes are set apart from the text and their purpose is clearly identified. For example:

NOTE: This is a box designated for a note. It contains information that it is set apart so as to catch your eye. Tips and cautions are handled similarly and labelled accordingly.

Chapter 1: Product overview

The NetBorder Call Analyzer Release 2.0.0 for Windows (XP and above) provides the flexibility needed to support a broad range of applications that require Call Progress Analysis (CPA) across all supported technologies and protocols.

Call Progress Analysis is the process of detecting pre-connect information about failed outbound call attempts and the destination party's media type for connected outbound calls. CPA is particularly important in the Enterprise and IP Contact Center, where reduced response times and CPA accuracy directly translate into increased productivity, reduced costs, and improved customer satisfaction and retention.

This chapter contains:

Introduction to Call Progress Analysis

Strictly speaking, Call Progress Analysis (CPA) is the automated determination by a piece of telecommunications equipment as to the result of dialing a number. For example, the result of the analysis might be a busy tone, an answered call, ringing at the other end but no answer after a preset number of rings or after a specified duration of time, and so on. The analysis involves detecting the various call progress tones generated by the telephone network as the call is put through.

Call progress tones

In telephony, call progress tones are audible tones sent from the *Public Switched Telephone Network (PSTN)* or a *Private Branch Exchange (PBX)* to calling parties to indicate the status of phone calls.

The four most common call progress tones are:

- dial tone
- busy tone
- audible ringback
- reorder tone (fast busy tone).

Special Information Tones (SITs)

In addition to the standard tones above, CPA can be used to detect *Special Information Tones (SITs)*. The special information tone is actually a series of three precise, sequential audio tones, which indicate that the callee cannot be reached. For example, a SIT tone alerts the caller that a recorded announcement will follow, explaining the failed call attempt to the caller. Automated dialing equipment, on the other hand, determine the reason for the failed call based on the frequency and duration of the tone itself.

The process

CPA involves the following three crucial steps:

1. **Placing the call.**

   This is typically achieved by using an automated (predictive) *dialer* . Frequently, this dialer is a server that has been preconfigured to make calls automatically; for example, to a list of customers.

2. **Performing Call Progress Analysis.**

   As the call is being established, a Netborder Call Analyzer “listens on the line” and determines “who” is answering the call. Is it a person? Is it a voice-mail greeting? Is it a busy tone? Is it a telephone company operator (if the number is de-listed, for example)?

3. **Applying a treatment to the call, depending on NCA results.**

   For example, in the Contact Center, should the CPA determine that the call was answered by a human, the call is transferred to an available agent. Or, should the CPA determine that the call was answered by voice mail, a pre-recorded message is played, to inform the customer of the call, for instance.

Benefits

Out of these operations, two very important factors emerge, which directly impact the success and efficiency of outbound dialing operations:

1. **CPA accuracy**

   If the Netborder Call Analyzer determines that a call has been answered by a person, while in fact it has encountered a voice-mail greeting, the agent receiving the call will spend time handling the call before he or she once again becomes available to answer another call. While this timeframe may be short (under a minute or so), it translates into a considerable loss of productivity when accumulated over time and in large volume call centers.

   Similarly, should the Netborder Call Analyzer determine that an answering machine has picked up the call, when in fact a person has answered, the Contact Center operator may face stiff fines for contravening regulations.

2. **Response time**

   If it takes too long to connect to an agent, there is an increased risk of not being able to deliver the service at all, whether the Netborder Call Analyzer is accurate or not. Customers may hang up on the call (the infamous “dead air”), or the Contact Center operator may face penalties for non-compliance of nuisance call regulations.

Introducing the statistical model

With the NetBorder Call Analyzer, call progress analysis is achieved through the use of statistical models based on neural networks to evaluate the potential outcome of an outbound call attempt. The result is an intelligent machine that “learns” the behaviour that represent the different conditions representing call progress events.

This method provides NCA results with far superior accuracy and flexibility compared to traditional approaches, which tend to rely on heuristics or “rules of thumb”. These statistic-based results provide:

- **Improved accuracy and response time.** See Benefits above.
- **More resilience against various network conditions.**

  This approach provides superior robustness against volume variations, background noise and other network conditions. Since Contact Centers are starting to use automated dialing for a much different set of applications (callback, automated notification, proactive campaigns), as opposed to a single application, the CPA platform must allow for per-call optimization of calling parameters.

- **Dynamic operations.**

  With the statistical approach, it is possible for the automated dialing application to select the particular operation point on a per-call basis. This means that the accuracy versus response time trade-off is locally optimized to provide the best possible results.

With its unique call progress analysis capabilities, the NetBorder Call Analyzer provides superior performance and optimal use of resources for IP-based outbound voice solutions.

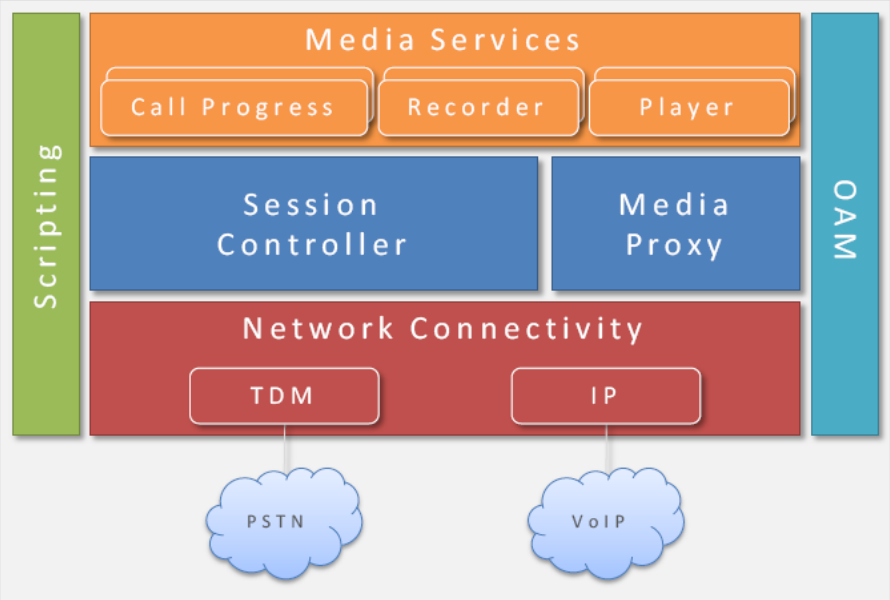
Product description

The NetBorder Call Analyzer is a component of Sangoma’s NetBorder Software Suite.

NetBorder Software Suite

The NetBorder Software Suite is a VoIP Session and Media Controller, a new class of product for VoIP deployments in the enterprise. NetBorder is comprised of four major software subsystems:

- Session Controller
- Media Proxy
- Media Services
- Network Connectivity.



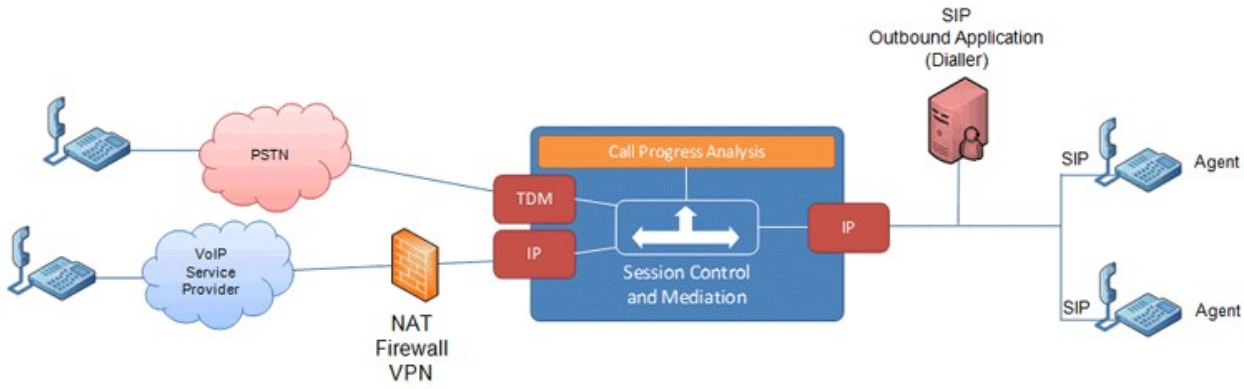
Each software subsystem has its own *OAM (Operations, Administration, and Maintenance)* components, for increased flexibility and manageability.

In NetBorder, these software subsystems combine to provide the necessary network connectivity, signaling proxy, signaling harmonization, media processing, media conversion and call control scripting, to remove the complexity from SIP applications.

NetBorder Call Analyzer

The NetBorder Call Analyzer is a specialized component of the Media Services subsystem of the NetBorder Software Suite.

The NetBorder Call Analyzer implements unique pattern-recognition algorithms to provide superior accuracy in distinguishing live responses from voice mail systems or other devices and signals. The technology significantly increases agent productivity, reduces connection time to callers, and streamlines the outbound calling deployment process.



Note that the application provides connectivity to both legacy and IP networks, and routes both inbound and outbound traffic through a single element.

This simplified approach, and particularly a process based on statistical models and SIP-based integration, permits the NetBorder Call Analyzer to deliver many improvements to the call progress analysis process.

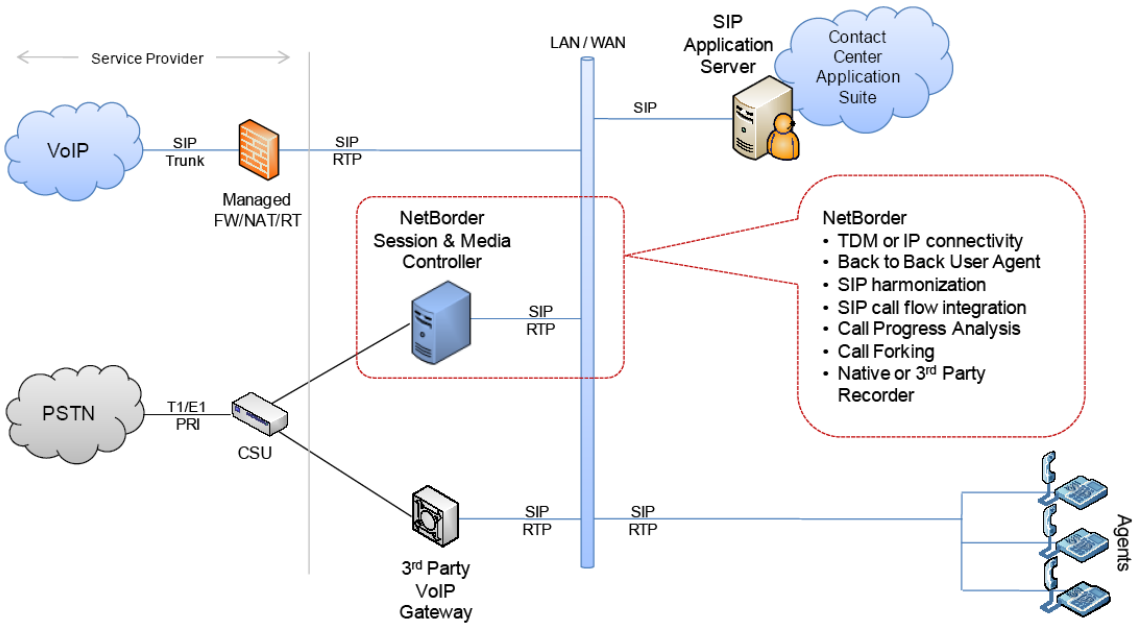
End of greeting detection

In order to further improve the productivity of a call center, NetBorder Call Analyzer allows the detection of the end of a greeting prompt when an answering machine is reached.

This optional feature allows the dialer application to leave a fully recorded, intellegible and clear notification message to the called party.

System architecture

Sangoma uses an innovative approach based on software platforms and SIP. The figure below depicts the architecture of a typical IP Contact Center.



On the left side of the diagram are the network interconnections of the Contact Center. These network interconnections may be delivered over traditional interfaces (T1/E1) or via SIP trunking. For the delivery of SIP services, the SIP trunk interconnects directly on the customer LAN/WAN. For TDM traffic, VoIP gateways are used to convert to SIP/RTP.

On the right side of the diagram are depicted the various users and applications:

- Application Server(s), delivering a complete suite of Contact Center applications (such as ACD, Dialer, IVR).
- SIP Agents, which are registered (as part of the SIP protocol) to the Application Server. Note that with VoIP, these agents can be located almost anywhere.
- NetBorder Suite (shown inside the dotted line), providing the necessary Session and Media Control mediation to deliver the complex call control required in Contact Center applications.

Chapter 2: Installation

This section describes how to install (and uninstall) the NetBorder Call Analyzer software.

For information on how to start using the service once the NetBorder Call Analyzer has been installed successfully, see Chapter 3: Getting started .

This chapter contains the following topics:

System requirements

Before you begin the installation process, please refer to the *Release Notes* to ensure compatibility with third party software and hardware.

Installing the software

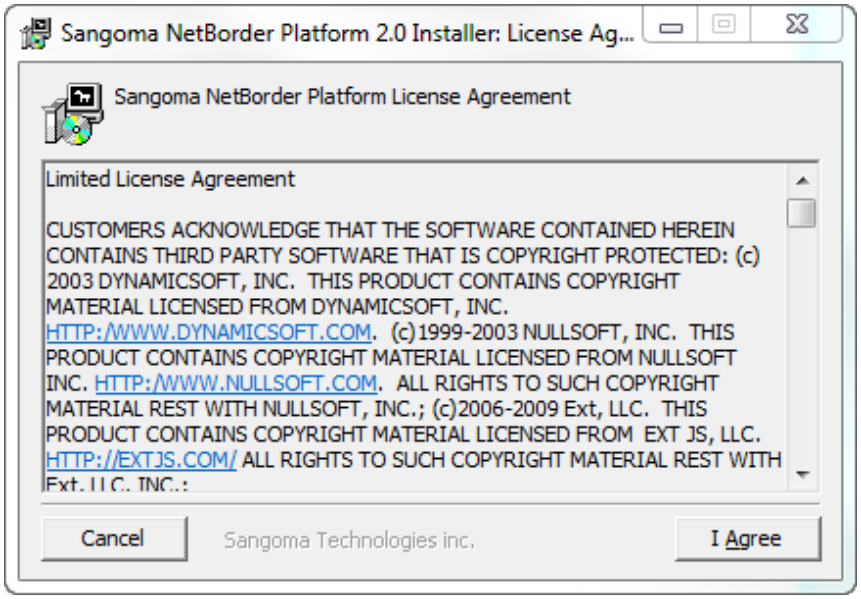
The NetBorder Call Analyzer software installer is available from the Sangoma website. If you have not been provided with the necessary URI, contact Sangoma at the following e-mail address:

- techdesk@sangoma.com

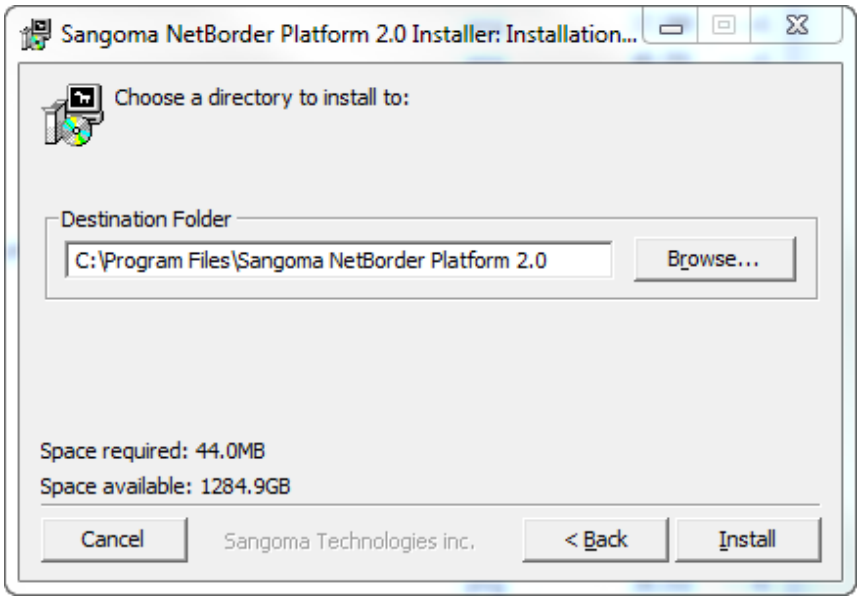
You must download this installer to your system before proceeding with the installation. Before starting the installation, exit all other applications.

To install the NetBorder Call Analyzer software on Windows®:

1. To begin the installation, double-click the following executable file:
   - *SangomaNetBorderPlatform.2.X.X.Setup.exe*
2. Carefully read the licensing terms for the software. If you agree to the terms, click **I Agree**.



   If you click **Cancel**, the installation will abort. The NetBorder Call Analyzer software will not be installed.
3. Set the home directory for the installation.



Do **one** of the following:

- Use the default folder (C:\Program Files\Sangoma NetBorder Platform 2.0\).
- Enter a new path.
  - Click **Browse...** to select a different directory.

4. Click **Install** to complete the installation process.

   The required directories, subdirectories and files are installed on your system. For a complete list, see Directories and files installed by the software below.

To install the NetBorder Call Analyzer software on Linux

NetBorder Call Analyzer is distributed on supported Linux distributions using the **rpm** format. To install:

rpm -i netborder-call-analyzer-2.<version>.<arch>.rpm

The NetBorder Call Analyzer software will be installed under the following directory:

/opt/Sangoma_NetBorderCallAnalyzer

This directory will be referred to as *[NETBORDER_INSTALLDIR]* throughout this document when the OS in used is Linux.

Directories and files installed by the software

Below is a list of the directories, subdirectories and files copied to the system during the installation process.

Note that in the table below, [*NETBORDER_INSTALLDIR*] is used to indicate the root folder of the installation (for example, *C:\Program Files\Sangoma NetBorder Platform 2.0*).

List of Files Installed by the Application Software	
Folders and Files	***Description***
[NETBORDER_INSTALLDIR]\	The root directory of the installation folder contains one file, the uninstall executable (*uninstall.exe*).
[NETBORDER_INSTALLDIR]\.original_config\	The *\.original_config* folder contains the original configuration files and python files (which were copied to your system during installation). This folder serves as a backup of the original configuration should you later encounter a problem. This folder also contains the original license TXT file, which lists information about the original license, including the version number of the software and the expiry date of the license.
[NETBORDER_INSTALLDIR]\bin\	The *\bin* folder contains the executable files (EXE) required to run the software, as well as the DLL (Dynamic Link Library) files, which contain code that is called upon when needed by the executable files.
[NETBORDER_INSTALLDIR]\config\	

List of Files Installed by the Application Software	
Folders and Files	***Description***
call-analyzer-service.properties call-analyzer-engine.properties call-analyzer-logging.properties call-analyzer-engine-logging.properties CallAnalyzerAsOutboundProxy.call-properties CallAnalyzerGenesysOCS.call-properties call-analyzer-license.txt call-analyzer-license.txt.sig	The *config* folder contains the main configuration files: *call-analyzer-engine.properties* and *call-analyzer-service.properties*. These files may be opened, modified and saved using any text editor. They may also be copied and replicated elsewhere. This folder also contains the *call-analyzer-engine-logging.properties* and *call-analyzer-logging.properties* used to configure the logging properties. You will also find the Netborder application class properties files: •CallAnalyzerAsOutboundProxy.call-properties •CallAnalyzerGenesysOCS.call-properties As well as the current license files. The *license.txt* file lists information about the existing license, including the version number of the software and the expiry date of the license.
[NETBORDER_INSTALLDIR]\data\tone-db	The \tone-db folder contains the tone frequency information stored in XML files for all the countries supported.
[NETBORDER_INSTALLDIR]\doc\	
cpa_release_notes.pdf cpa_user_guide.pdf	The *doc* folder contains relevant documentation, including the following: • Release Notes • This User's Guide.
[NETBORDER_INSTALLDIR]\logs\	
Cpa-stats.csv * call-analyser-service.out call-analyser-engine.out *Created on first call made via NCA.	The *logs* folder contains all the log files, including cpa-stats.csv, which includes call statistics. See The Cpa-stats.csv file below. This folder will be empty immediately after installation.

List of Files Installed by the Application Software	
Folders and Files	***Description***
[NETBORDER_INSTALLDIR]\logs\call-logs\	The *logs**call-logs* folder contains files reporting activity for each call (“call logs”). There is one file by call, and files are output following a directory hierarchy based on the current date and time. When recording is enabled (see Recording media on page 65), the recordings in .wav format (G.711 mulaw/Alaw codecs) are output in the same location than the call-logs files.

The Cpa-stats.csv file

When a call with call progress analysis is completed, details about the call are logged on a single line in a concise way using comma separated values.

By default, this file is a 'DailyRollingFile', i.e.: it will automatically rename itself at the end of the day by appending the date to the file name. Only the last 10 files are kept on the disk, older one being discarded. See Appendix C: Logging configuration for description of additional functionality.

The logged informations are described in the table below.

Cpa-stats.csv field description	
Field	***Description***
NetBorder Call-ID	The unique identifier used by NetBorder Call Analyzer to identify the call
Call Date	The date the call was initiated. (YYYY-MM-DD)
Reference ID	The optional campaign specific call reference ID provided by the customer via the X-Netborder-Cpa-Reference-ID SIP header. (See page 51 for more information)
Campaign Name	The optional campaign name provided by the customer via the X-Netborder-Cpa-Campaign-Name SIP header. The combination of the campaign name and reference ID above could be used as a unique ID for a call. (See page 51 for more information)
Phone Number	The dialed phone number
NCA result	The result of the CPA analysis
Time Dialed	The timestamp when NetBorder Call Analyzer received the initial SIP INVITE request.

Cpa-stats.csv field description	
Field	***Description***
Time Connected	The timestamp when NetBorder Call Analyzer received the SIP 200 OK response from the callee. Empty if the callee never connected.
Time CPA completed	The timestamp when the CPA analysis provided a final outcome.
Time queued	Used only when the initial INVITE received by NetBorder Call Analyzer did not contains SDP information. In this case, it is the timestamp when NetBorder Call Analyzer sent the 200 OK with NCA result to the dialer. This field remains empty if the initial INVITE contains SDP information.
Time connected to agent	The timestamp when the agent connected with the callee. I.E. when the dialer sent its ACK following the 200 OK.

Obtaining and installing the license file

The NetBorder Call Analyzer default license provides customers with the ability to place one call at a time. Attempts to place simultaneous calls will be rejected with a 603 Decline SIP response

- Look for *603 Decline (Licensing capacity has been reached)* in file: *[NETBORDER_INSTALLDIR]/* logs/call-analyzer-service.out if you get 603 Decline responses to confirm it is a license capacity issue.

To obtain a **full license** (host-locked) , obtain the *MAC (Media Access Control)* address of the system and use the Installation ID that came with the software to generate a license file. Please follow this URL:

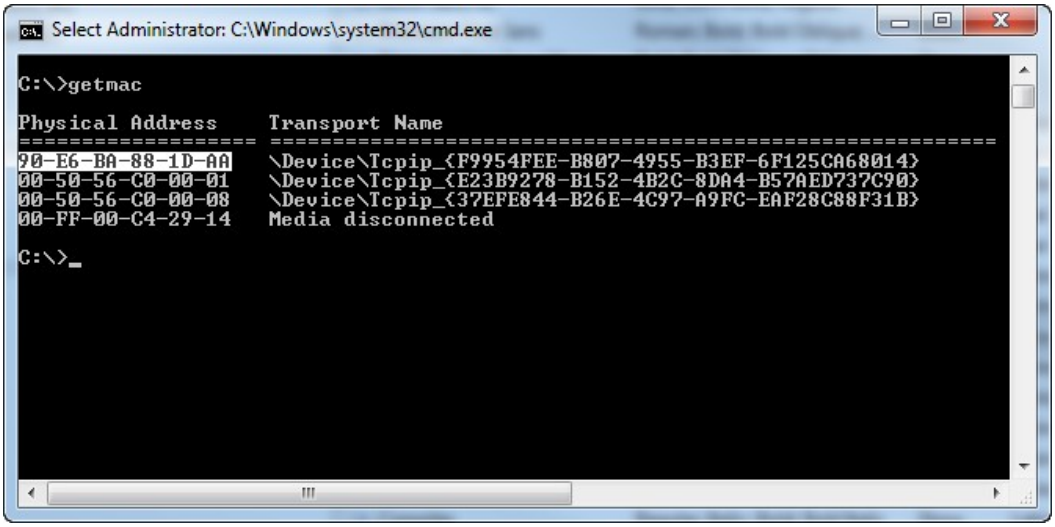
- http://www.sangoma.com/support/register_netborder_software.html

To obtain the physical address of the Ethernet adapter (on Windows Vista and up):

1. Start a command prompt and execute the following command:
   - `getmac`

TIP: To start a command prompt, select **Run** from the Start Menu. Type " `cmd` " in the textbox, and click **OK**.

2. Look for the *Physical Address* item. For example, it should look something like this:
   - `90-E6-BA-88-1D-AA`



To obtain the physical address of the Ethernet adapter (on Windows XP and 2003):

1. Start a command prompt and execute the following command:
   - `ipconfig /all`

TIP: To start a command prompt, select **Run** from the Start Menu. Type " `cmd` " in the textbox, and click **OK**.

2. Look for the *Physical Address* item. For example, it should look something like this:
   - `90-E6-BA-88-1D-AA`

To obtain the physical address of the Ethernet adapter (on Linux distributions):

1. Execute the following command in a shell:
   - `ifconfig`
2. Look for the *Hwaddr* field value for a network adapter that is permanent on your server, usually *eth0*.

3. The *Hwaddr* value is shown using the XX:XX:XX:XX:XX:XX notation. Upon registration of the software, the license provided will show the physical address (MAC address) using the Windows notation (XX-XX-XX-XX-XX-XX): this is expected.

To install the license files:

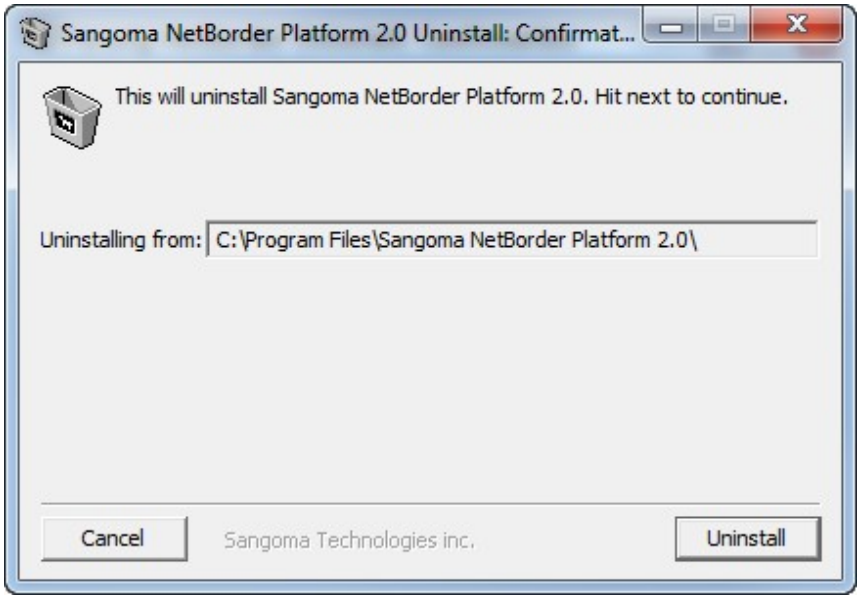
1. Upon filling the online form, you will be able to download your new license files in a .zip file:
   - *call-analyzer-license.txt*
   - *call-analyzer-license.txt.sig*
2. Once you have received the license files, copy the files (in effect, you will be replacing the temporary license) to the following folder:
   - *[NETBORDER_INSTALLDIR]\config*

     where *[NETBORDER_INSTALLDIR]* is the root folder of the installation (for example, *C:\Program Files\Sangoma NetBorder Platform 2.0\config*).
3. Start the Sangoma NetBorder Call Analyzer service. See Starting the service on page 29.

Uninstalling the software

To uninstall the NetBorder Call Analyzer software on Windows:

1. Launch the uninstaller from the Start Menu: select **Programs** > **Sangoma NetBorder Platform 2.0** > **Uninstall** . The following window appears:



2. Click **Uninstall**.

   The NetBorder Call Analyzer software is removed from your system.

NOTE: You can also remove the software using the **Add/Remove Program Manager** found in the **Control Panel**. Alternatively, you can run the following command from a Windows command line: `[ NETBORDER_INSTALLDIR ]\uninst.exe` (where `[ NETBORDER_INSTALLDIR ]` is the root folder of the installation).

To uninstall the NetBorder Call Analyzer software on Linux:

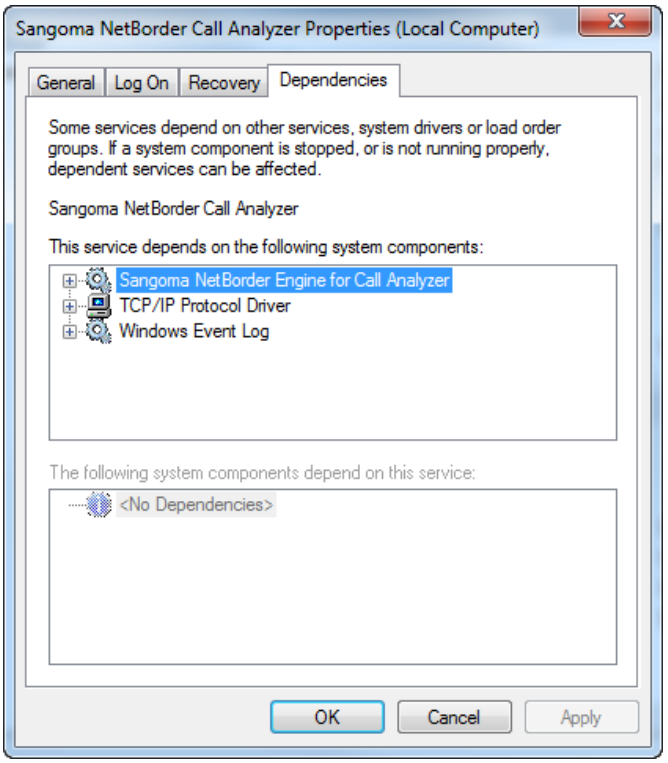
1. Make sure you backed up any file that you manually edited under [NETBORDER_INSTALLDIR]/config
2. Execute the following command in a shell:
   - *rpm -e netborder-call-analyzer*

Validating the installation

Once you have installed the NetBorder Call Analyzer software, you should validate the successful installation by starting the application.

Note that two services/processes makes up NetBorder Call Analyzer:

- **Sangoma NetBorder Call Analyzer:** Provides Call Progress Analysis Services to the NetBorder framework. It is the entry point where SIP INVITE requests must be sent (default is UDP port 5062)
  - Process name: *netborder-call-analyzer-service*
- **Sangoma NetBorder Engine For Call Analyzer:** Internal service that hosts the Engine performing the analysis on the media stream.
  - Process name: *netborder-call-analyzer-engine*



Sangoma NetBorder Call Analyzer service dependencies on Windows

To start all the services required to run the NetBorder Call Analyzer on Windows, you only need to start the *Sangoma NetBorder Call Analyzer* service.

Starting the service

To start the NetBorder Call Analyzer on Windows:

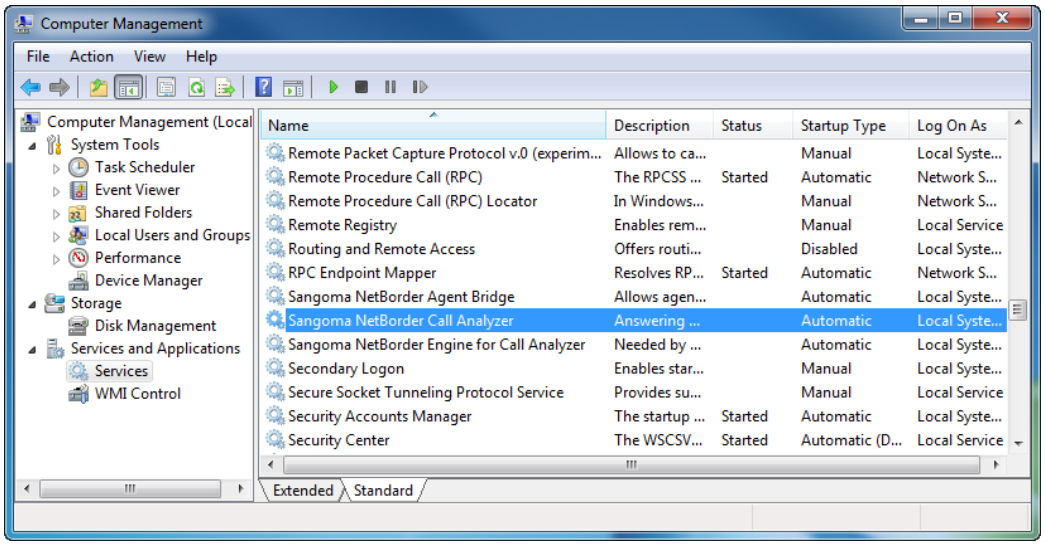
1. Open the Services list, as follows:

   a) Right-click **My Computer**, and click **Manage**.

   If you don't have a My Computer icon on your desktop, click the **Start** button and look for **My Computer** listed on the right side of the Start Menu.

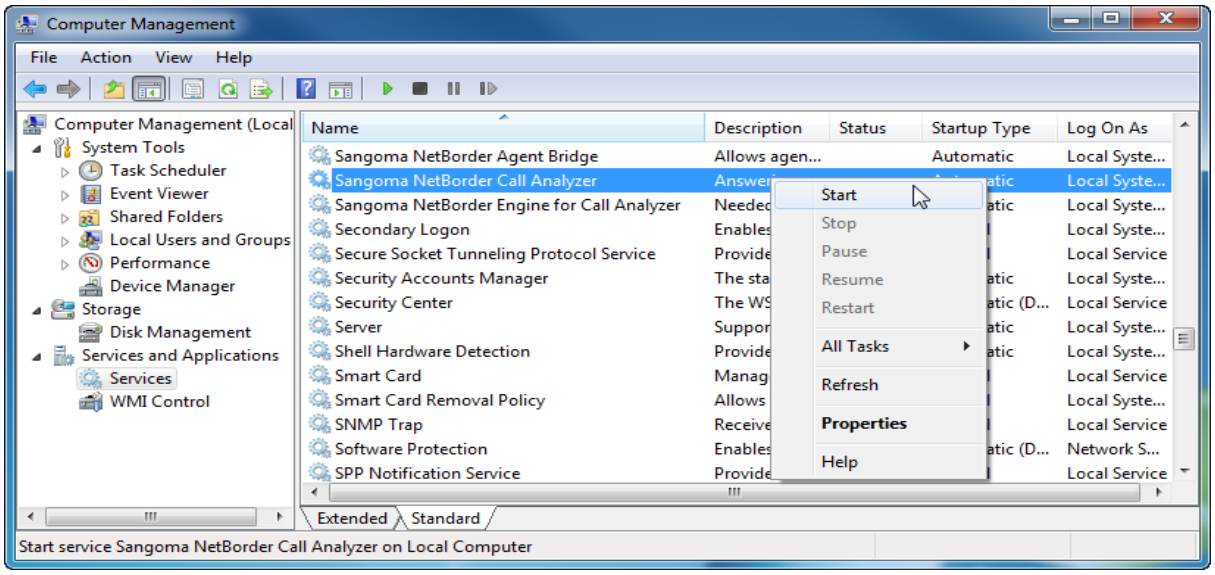
   Alternatively, you can access the Services list from the **Control Panel**: double-click **Administrative Tools** and then double-click **Services**.

   b) In the left pane, select "Services" under the "Services and Applications" folder.

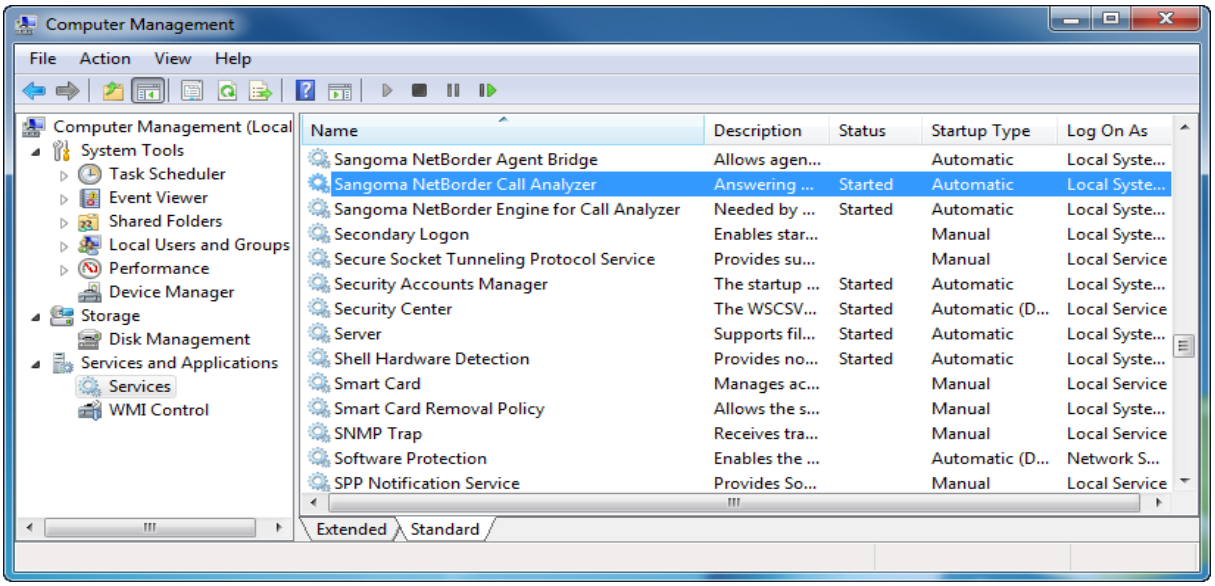


1. In the Services list, right-click "Sangoma NetBorder Call Analyzer" and select **Start** from the context menu.

If the services are already running, you will see their Status as “Started”. In this case, the option “Start” will not be accessible from the right-click action menu.



The Sangoma NetBorder Call Analyzer and related services will start.



You are now ready to place a call through your IP network. Turn to Chapter 3: Getting started. However, if you encountered an error, see What to do if the service fails to start on page 32.

To start the NetBorder Call Analyzer on Linux

The package name for NetBorder Call Analyzer on Linux is:

- *netborder-call-analyzer*

The installer also creates a service of the same name, so the commands to control the service are:

- *service netborder-call-analyzer {start|stop|status}*

The “start” command will launch all necessary processes for the service, “stop” will terminate them and “status” will tell if the service is started or not.

Automatic service startup on Linux

The *netborder-call-analyzer* service is setup to start up automatically on server boot-up: To change that behaviour, execute the following command in the shell:

- *chkconfig netborder-call-analyzer off*

To restore automatic startup, use:

- *chkconfig netborder-call-analyzer on*

What to do if the service fails to start

If the service fails to start, try the diagnostic steps below:

1. **Windows:** Check the Windows Event Viewer to see if any events of level FATAL, ERROR or WARN have been detected with the software. See Viewing logs of high-level events on page 67.
2. **Linux:** The syslog service is used to output major errors and warnings, under the “netborder-call-analyzer” identity. By default, the log messages will be output in the */var/log/messages* file.
3. Check the log files for errors. See Viewing logs of low-level events on page 69.

For further assistance, see Chapter 6: Troubleshooting.

Chapter 3: Getting started

Once you have installed the NetBorder Call Analyzer and started the service, you should place an outbound call to verify that a call is successfully put through, and that call progress analysis is functioning and returning the expected results.

This chapter contains the following topics:

This chapter assumes that the NetBorder Call Analyzer software has been successfully installed on your system. For installation procedures, please read Chapter 2: Installation.

Prerequisites to making a test SIP call

To make a test SIP call, you will need the following:

- PC-based softphone, which provides the same functionality as a typical handset and integrates with other multi-service applications such as web browsing and instant messaging; examples include the Kapanga Softphone and the CounterPath X-Lite.

NOTE: If you do not have a softphone, both of the products listed above are available for download at the URIs provided. For the purposes of this test, you will be making a SIP to SIP call; therefore, you will need two softphones, one on your system, and another installed on a different system (the callee).

- Full-duplex sound card.
- Speakers and microphone, or a headset.
- Connection to an IP network.

You will also need to configure the softphone to use NetBorder Call Analyzer as an outbound proxy. See Configuring as an outbound proxy on page 35.

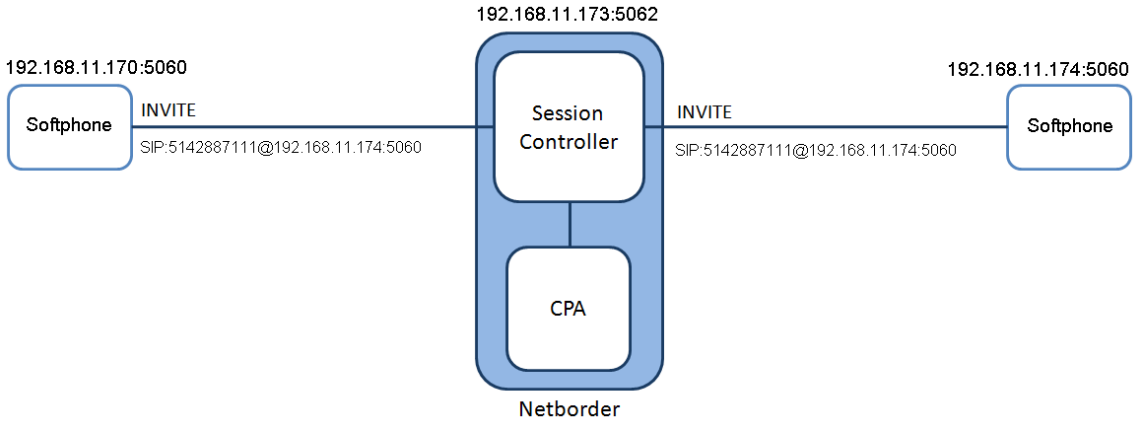
Configuring as an outbound proxy

By default, the NetBorder Call Analyzer is configured to serve as an outbound proxy. Before making your test call, you will need to configure your softphone to route calls through NetBorder.

NOTE: If the *User Agent Client (UAC)* or caller cannot use NetBorder as an outbound SIP proxy, a relay server may be used. For more information, see Configuring NetBorder to be used with Genesys SIP Server (ie relay server) on page 60.

The following steps feature X-Lite. Configuration of other softphones will follow the same general procedures.

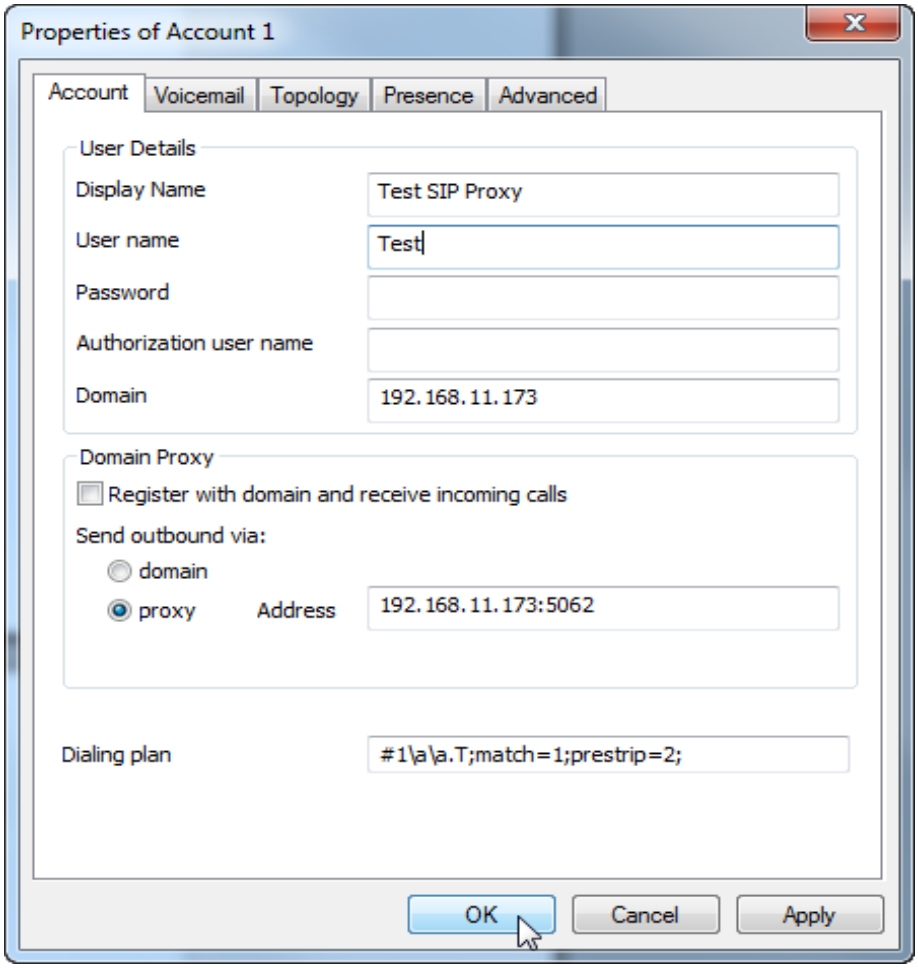
The step to use the NetBorder Call Analyzer is to make sure you can use it in your own environment. To do so, the setup illustrated below should be used. Note that the caller softphone, on your left, is located on its own computer with a different IP address than the NCA, in the center and the called softphone, on your right. It would also be possible to use the same computer for the caller softphone and the NCA, but the callee should be on its own computer.



To set up proxy configuration parameters:

1. Access the softphone's SIP Account settings menu .

2. Depending on the softphone you are using (for example, X-Lite), add a new profile.



3. Enter the required SIP Proxy settings:
   - **Domain:** Used to form all destination URIs when a telephone number is entered without specifying the domain explicitly.
   - **Proxy usage mode:** “Outbound Proxy” or “Strict Outbound Proxy”, depending on the softphone used.
   - **Outbound proxy Address:** The full SIP URI of the outbound proxy. This URI should consist of the “`sip:`” prefix, followed by the IP address of the proxy, and the port number (“5062” for NetBorder Call Analyzer).

     For example: sip:192.168.11.173:5062.

CAUTION: When using “5062” as the listening port for NetBorder, it is important not to run any other application that may cause conflict on that port (such as a SIP phone). In general, SIP phones use “5060” as the default listening port.

By configuring the softphone to use an outbound proxy, the softphone will initiate all its SIP communication through the outbound proxy; that is, the callee's SIP URI will be processed by NetBorder. In other words, all SIP requests and responses will be sent through NetBorder.

Making a call without CPA

To further verify the NetBorder Call Analyzer installation, you should place an outbound call to ensure that it is put through and you receive valid results.

Based on the SIP INVITE arguments that are received by NetBorder, the call progress analysis can be activated or deactivated. If no arguments are present, the call is placed without call progress analysis.

Once an outbound call has been placed, NetBorder receives the request and re-originates the outbound call towards the VoIP provider network. Thus, NetBorder is acting as a User Agent Server *(UAS)*. To answer the incoming SIP request, NetBorder will act as a User Agent Client *(UAC)*, regenerate the SIP request, and send it over the network.

Again, the following steps feature X-Lite. Other softphones will follow the same general procedures.

To initiate an outbound call:

1. Start your softphone application.
2. Place a call using the softphone. Enter the phone number in the "Call To" field using dotted notation, as follows:

   `sip:[phone number or extension]@[IP address]:[port]`

   For example: sip:1024@192.168.11.103:5061.

   In the example above, X-Lite is being used to call the telephone extension "1024", reachable via the PSTN gateway at 192.168.11.103:5061.

3. Place the call. If you are using X-Lite, press **Enter** or click the green telephone icon in the left side of the application.
4. Verify that you hear a ringing tone and ultimately, once the call is connected, audio on the other end.
5. Interact with the application to verify that audio is coming through on both ends of the call.

NOTE: If you fail to hear audio, your firewall may be preventing calls from being put through. Also, if the call is established successfully and no audio is heard, the firewall may be blocking the media (RTP) packets. Try disabling your firewall. If you continue to experience problems, see What to do if a call is not connected on page 44.

6. When finished, hang up.

   If you are using X-Lite, click the red telephone icon on the right side of the application.

If you receive an error message, such as a “Decline” message from the softphone application, see What to do if a call is not connected on page 44.

7. If you wish, you can view SIP messages related to the call to verify that you have received the expected results. The call-log files are located in the following directory:

   - *[NETBORDER_INSTALLDIR]\logs\call-logs*

     where *[NETBORDER_INSTALLDIR]* is the root folder of the installation (for example, *C:\Program Files\Sangoma NetBorder Platform 2.0\logs\call-logs).*

   By default, call-log files are saved in subdirectories based on date and time.

Making a call with CPA

Now place an outbound call with CPA on to verify that call progress analysis is functioning properly and returning useful results. The CPA scenario is triggered by placing “ `;cpd=on` ” in the SIP Invite. Therefore, this process is very similar to the previous procedure, except that you add “ `;cpd=on` ” to the phone number you are dialing.

To initiate an outbound call with CPA:

1. Start your softphone application.
2. Place a call using the softphone. Enter the phone number in the “Call To” field using dotted notation, as follows:

   `sip:[phone number or extension]@[IP address]:[port]`

   For example: sip:1024@192.168.11.103:5061.
3. Append the following to the phone number you entered: “`;cpd=on`”.

   For example: sip:1024@192.168.11.103:5061;cpd=on.
4. Place the call.

5. Verify that you hear a ringing tone and ultimately, once the call is connected, audio on the other end.
6. Interact with the application to verify that audio is coming through on both ends of the call.
7. When finished, hang up.

   If you are using X-Lite, click the red telephone icon on the right side of the application.

   If you receive an error message, such as a “Decline” message from the softphone application, see What to do if a call is not connected on page 44.
8. Now take a look at the call-log files to verify the results of the call and particularly of the call progress analysis.

Do the following:

a) Open the call-log files located in the following directory:

- *[NETBORDER_INSTALLDIR]\logs\call-logs*

  where *[NETBORDER_INSTALLDIR]* is the root folder of the installation (for example, *C:\Program Files\Sangoma NetBorder Platform 2.0\logs\call-logs).*

By default, call-log files are saved in subdirectories based on date and time.

a) Take a look at both the SIP request and SIP response messages. Conduct a search for the following:

- "CPD-Result" in the .log file: For the result of the call progress analysis (for example, "`CPD-Result: Voice`" in a SIP response header).
- "CPA" in the .analyzer-engine.log file: To locate information such as the degree of certainty (expressed in decimal percentages) of the NCA result. For example:

  ```
  CPA_HUMAN=0.710736
  CPA_MACHINE=0.150314
  CPA_FAX=0.13895
  ```

  For more information on SIP messages, including the components of a SIP message (for example, the CPD-Result header field), see Chapter 4: Call flow fundamentals . For more information on logging, see Setting up logging on page 66 .

What to do if a call is not connected

If a call is not connected, and/or you receive a message from the softphone application informing you that the call has been rejected or declined, try one or more of the diagnostic steps below:

1. Ensure that all related Sangoma NetBorder services are running (not **only** the Sangoma NetBorder Call Analyzer service is started **but** the Sangoma NetBorder Engine For Call Analyzer). See Starting the service on page 29.
2. Check the Windows Event Viewer to see if any events of level FATAL, WARN or ERROR have been detected with the software. See Viewing logs of high-level events on page 67.
3. Check the log files for errors. See Viewing logs of low-level events on page 69.
4. Check the call-logs files and specifically any SIP response error codes. Search for the text string, “sip.message”.

   For example, this sample response message indicates a “603 Decline” SIP response code, as well as a CPD-Result of “Reject” in the SIP call-log:

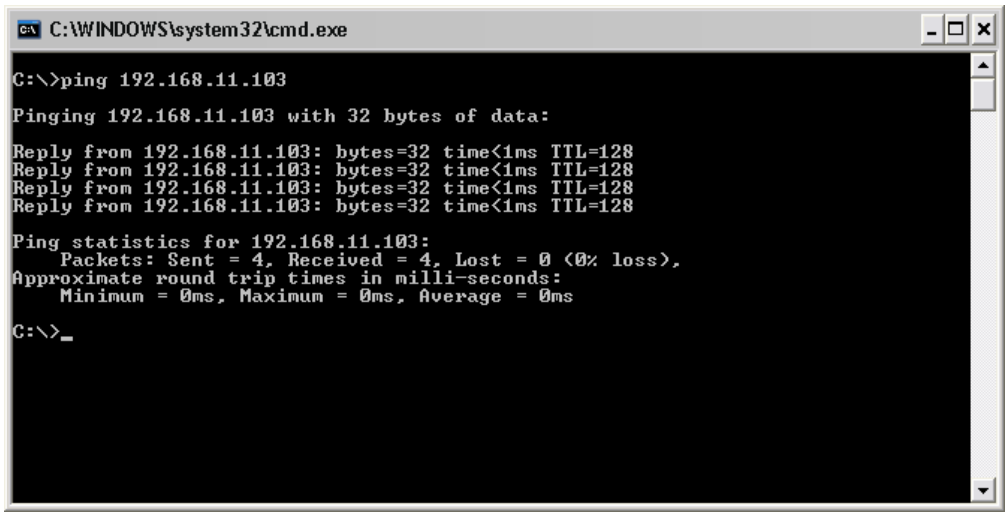
   ```
   SIP/2.0 603 Decline
   Via: SIP/2.0/UDP
   127.0.0.1;branch=z9hG4bKc0a80bef0000002f4713aafa00000ce300000014;
   rport=5060;
   received=192.168.11.239
   To: <sip:1024@192.168.11.103:5061;cpd=on>;tag=76597b4a
   From: "unknown"<sip:127.0.0.1>;tag=45b6e446a8
   Call-ID: 6C9AD1EF4DCB4C88AC4E9C53E617609C0xc0a80bef
   CSeq: 1 INVITE
   Content-Length: 0
   CPD-Result: Reject
   ```

5. Be sure the IP address you are calling is valid. Use the `ping` command to verify IP-level connectivity to the other TCP/IP user.

Start a command prompt and execute the following command:

- `ping <IP address>`

  where `<IP address>` is the address of the callee or the person you are dialing; for example, 192.168.11.103.



TIP: To start a command prompt, select **Run** from the Start Menu. Type "`cmd`" in the textbox, and click **OK**.

1. Check your connections. Ensure that the IP connectivity to the host system is functioning. If necessary, check physical connections.

For further assistance, see Chapter 6: Troubleshooting

Chapter 4: Call flow fundamentals

This chapter describes the structure of SIP messages, including custom header fields unique to the NetBorder Call Analyzer. It also provides a sample call flow accompanied by a detailed explanation.

This chapter contains the following topics:

SIP transactions and dialogs

A SIP signaling session between two *User Agents (UAs)* is composed of one or more SIP transactions. A SIP transaction occurs between a *User Agent Client* (UAC) and a *User Agent Server* (UAS). It might involve one or more intermediate SIP servers such as a proxy or redirect server. A SIP transaction comprises all messages that begin with the SIP request initiated from the UAC, until a final response (a non-1XX or non-provisional) is received from the UAS.

Typically, a SIP transaction comprises a SIP request message followed by one or more SIP response messages.

SIP messages

Each SIP transaction consists of a request that invokes a particular method, or function, on the server, and at least one response.

SIP requests

SIP requests are messages that are sent from client to server to invoke a SIP operation. RFC 3261 defines six requests or methods that enable a User Agent or SIP proxy to locate users and initiate, modify, and tear down sessions:

- **INVITE:** An INVITE method indicates that the recipient user or service is invited to participate in a session. This method can also be used to modify the characteristics of a previously established session.
- **ACK:** An ACK request confirms that the UAC has received the final response to an INVITE request. ACK is used only with INVITE requests.
- **OPTIONS:** An OPTIONS request is used to query servers about their capabilities. If the UAS is capable of delivering a session to a user, it responds with its capability set.
- **BYE:** A BYE request signifies the termination of a previously established session.
- **CANCEL:** A CANCEL request allows UACs and network servers to cancel an in-progress request, such as an INVITE.

- **REGISTER:** A REGISTER request is used to register the current contact information.

In addition, RFC 3515 defines the **REFER** method. This SIP extension requests that the recipient REFER to a resource provided in the request. This method can be used to enable many applications, including call transfer.

SIP responses

A server sends a SIP response to a client to indicate the status of a SIP request that the client previously sent to the server. Specifically, the UAS or proxy server generates SIP responses in response to a SIP request that the UAC initiates.

SIP responses are numbered from 100 to 600. Appendix D lists common SIP responses you may encounter while using the NetBorder Call Analyzer. For a complete list, see Section 21 of RFC 3261.

SIP message structure

A SIP message consists of the following four main components:

- start-line
- one or more header fields
- empty line indicating the end of header fields
- optional message body.

Start-line

The start-line for a SIP request is known as the Request-Line. The start-line for a SIP response is known as the Status-Line.

The Request-Line specifies the SIP method, the Request-URI, and the SIP version. With NetBorder Call Analyzer applications, by default the Request-Line also indicates whether call progress analysis is requested.

The Status-Line describes the SIP version, the SIP response code, and an optional reason phrase. The reason phrase is a textual description of the 3-digit SIP response code.

SIP headers

A SIP message is composed of header fields that convey the signaling and routing information for the SIP network entities (User Agent, proxy, B2BUA, and so on). Each header field consists of a field name followed by a colon (:) and the field value. For a description of the key SIP headers, refer to Section 7.3 of RFC 3261.

CPD-Result

The 'CPD-Result' header is provided for legacy purposes. Please see the X-Netborder-Detailed-CPD-Result header below which contains an extended level of details over the 'CPD-Result' header.

X-Netborder-Detailed-CPD-Result

The SIP header 'X-Netborder-Detailed-CPD-Result' informs the end-user of the audio analysis result returned by the NetBorder Call Analyzer. See the X-Netborder-Detailed-CPD-Result Table for details on the different possible values for this SIP header.

DetailedCpdResult	*CpdResult*	*Sip Code*	*Explanation*
Pre-Connect Results			
'Busy'	'Busy'	486	The Netborder Call Analyzer has detected a busy tone or a “486 Busy” was received from the callee.
'Reorder'	'All-Trunks-Busy'	480	The Netborder Call Analyzer has detected a reorder tone or a “600 Busy Everywhere” was received from the callee.
'Reject'	'Reject'	See exp.	The callee returned an error code (4xx, 5xx or 5xx) that is not either: “404 Not Found”. Permanent Situation”, “408 Request Timeout”, “480 Temporary Unavailable”, “486 Busy Here” or “600 Busy Everywhere”.
'Reject - licensing capacity exceeded'	'Reject'	603	A “603 Decline” has been received because the licensing capacity is not sufficient to handle the number of simultaneous calls in the system. Contact a Sangoma salesperson to obtain a license with more capacity.
'Sit-Temporary'	'Sit-Reorder'	480	Netborder Call Analyzer has detected a Temporary SIT.[1]

'Sit-Permanent'	'Sit-Vacant'	404	Netborder Call Analyzer has detected a Permanent SIT.[2]
'Cancelled'	'Unknown'	N/A	The dialer sent a SIP CANCEL message before the NetBorder Call Analyzer completed its analysis. The dialer's behavior should be adapted so this situation NEVER happens. Use NetBorder Call Analyzer preconnect timeout mechanism to limit duration of dialing attempt.
'No-Answer'	'No-Answer'	408	"408 Request Timeout" received, or the timeout defined by the *app.nca.PreConnectTimeout.ms* (specified in the .call-properties file[3]) configuration parameter has been reached . See Appendix B .
Post-Connect Results			
'Unknown'	'Unknown'	200 OK	The Netborder Call Analyzer was unable to detect a known result from the callee audio before the post-connect timeout has been reached. If using *End Of Greeting Detection*: the End Of Greeting timeout has been reached before the end of the greeting could be detected.
'Reject'	'Reject'	200 OK	The caller or callee hung up before the Netborder Call Analyzer was able to identify the call.
'Answering Machine'	'Answering-Machine'	200 OK	The Netborder Call Analyzer has detected an answering machine.
'End of Greeting - Beep'	'Answering-Machine'	200 OK	The Netborder Call Analyzer has detected an answering machine followed by a 'beep'.
'End of Greeting - Silence'	'Answering-Machine'	200 OK	The Netborder Call Analyzer has detected an answering machine followed by silence.
'End of Greeting - Timeout'	'Answering-Machine'	200 OK	The Netborder Call Analyzer has detected an answering machine and could not detect what followed before a timeout was triggered. The result should be interpreted as an answering machine.
'Human'	'Voice'	200 OK	The Netborder Call Analyzer has detected a human voice.
'Fax'	'Fax'	200 OK	The Netborder Call Analyzer has detected a fax.
'???'	'???'	N/A	A problem was encountered by the Netborder Call Analyzer.

1 One of the following for North America: SIT NC, SIT RO
2 One of the following for North America: SIT IC, SIT IO, SIT VC
3 E.g.: CallAnalyzerAsOutboundProxy.call-properties

X-Netborder-Call-ID

When NetBorder Call Analyzer includes a CPD-Result header in a SIP response message, it also add the custom SIP header named “X-Netborder-Call-ID ” that contains the unique call-id that was used internally to process this call. It is the same ID that is used as filename for logs, recording and in cpa-stats file. It may reliably be used to cross-reference calls from NetBorder Call Analyzer logs to the dialer application logs or database entries.

It looks like this:

```
X-Netborder-Call-ID : 1257866196-453125-14693-23
```

X-Netborder-Cpa-Reference-ID

The NetBorder Call Analyzer may optionally use a custom SIP header named “X-Netborder-Cpa-Reference-ID” to associate any given string identifier with its own call identifier. The given identifier is expected to be unique in the scope of a campaign. If more than one campaign go through the same NetBorder Call Analyzer, you may want to use X-Netborder-Cpa-Campaign-Name SIP header to further identify the call.

The sole purpose of this reference is to help map NetBorder Call Analyzer logs with the caller UAC application logs, it is not used by NetBorder Call Analyzer except for adding it in the cpa-stats.csv file. (See section “The Cpa-stats.csv file“ on page 23).

In order to use this feature, add the following header to the initial INVITE message sent to NetBorder Call Analyzer:

```
X-Netborder-Cpa-Reference-ID: XXX-XXX
```

Where “XXX-XXX” is replaced by any string of your choice.

X-Netborder-Cpa-Campaign-Name

The NetBorder Call Analyzer may also optionally use a custom SIP header named “X-Netborder-Cpa-Campaign-Name“ to associate a call to a specific campaign name. The specified name has no meaning for NetBorder Call Analyzer, it is not used except for adding it in the cpa-stats.csv file. (See section “The Cpa-stats.csv file“ on page 23).

In order to use this feature, add the following header to the initial INVITE message sent to NetBorder Call Analyzer:

```
X-Netborder-Cpa-Campaign-Name: XXX-XXX
```

Where “XXX-XXX” is replaced by any string of your choice.

SDP body

The *SDP (Session Description Protocol)* body contains information about the message. The SDP body is optional. For a complete explanation of the SDP session description, see RFC 2327.

Sample SIP messages

Below are two sample messages: a SIP request message, followed by a SIP response message. Note the string “`cpd=on`” in the Request-Line. This indicates that call progress analysis is requested.

Sample SIP Request Message	
INVITE sip:1024@192.168.11.103:5062; cpd=on SIP/2.0	Request-Line
Via: SIP/2.0/UDP 192.168.11.156:5060;branch=z9hG4bK233E73631D7A9A90B05D8C5A02 983766;rport=5060 Max-Forwards: 70 Contact: <sip: Username@192.168.11.156: 5060;transport=udp> To: <sip: 1024@192.168.11.103 :5062> From: "Your Long Name"<sip: Username@defaultproxy :5060>;tag= 660A262 2E4A7F9B9A839A2B8B9381FD Call-ID: FC735409EBE21C394E69C4AD800FBF01@192.168.11.156:5062 CSeq: 1 INVITE Session-Expires: 1800;refresher=uac Content-Type: application/sdp Supported: timer, replaces User-Agent: Kapanga Softphone Desktop 1.00/2163e+1161886252_001372BDBCE2 Content-Length: 323	SIP message headers
	Blank line
V=0 o=Username 1190134451 1190750689 IN IP4 192.168.11.156 s=Kapanga [1190134451] c=IN IP4 192.168.11.156 t=0 0 m=audio 5562 RTP/AVP 8 0 101	SDP body in SIP message

a=rtpmap:8 pcma/8000 a=sendrecv a=silenceSupp:off - - - a=rtcp:5563 a=maxptime:20 a=ptime:20 a=rtpmap:0 pcmu/8000 a=rtpmap:101 telephone-event/8000 a=fmtp:101 0-15,36	

Sample SIP Response Message	
SIP/2.0 200 OK	Status (Response) Line
Via: SIP/2.0/UDP 192.168.11.156:5060;branch=z9hG4bK233E73631D7A9A90B05D8C5A029 83766;rport=5060 Contact: "Sangoma NetBorder"<sip:NetBorder@127.0.0.1:5062> To: <sip:1024@192.168.11.103:5062>;tag=0f7fe217 From: "Your Long Name" <sip:Username@defaultproxy : 5060>;tag=660A72622E4A7F9B9A839 A2B8B9381FD Call-ID: FC735409EBE21C394E69C4AD800FBF01@192.168.11.156:5062 CSeq: 1 INVITE Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, INFO Content-Type: application/sdp Content-Length: 236 CPD-Result: Voice X-Netborder-Detailed-CPD-Result: Human	SIP message headers
	Blank line
v=0 o=Paraxip-Tech 1190750737 1190750739 IN IP4 192.168.11.103 s=SIP Call c=IN IP4 192.168.11.103 t=0 0 m=audio 49152 RTP/AVP 0 101 a=fmtp:101 0-16 a=ptime:20 a=rtpmap:0 pcmu/8000 a=rtpmap:101 telephone-event/8000 a=sendrecv	SDP body in SIP 200 OK message

Notice the value of the CPD-Result header field. The NetBorder Call Analyzer has determined the result of the call analysis to be “Voice” (a person, as opposed to a machine, has evidently answered the call).

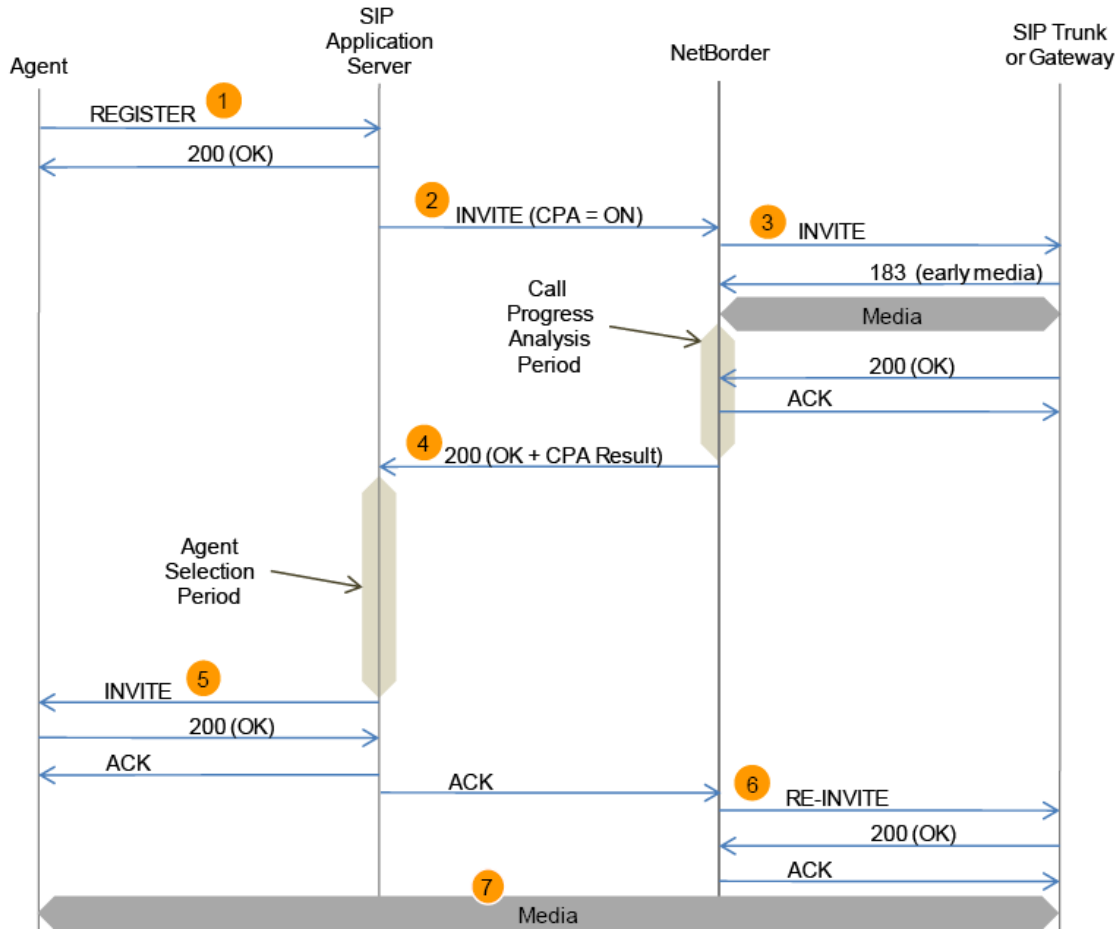
Sample call flow

Typically, a transaction commences when a participant is invited to a call.

In the Contact Center, the outbound dialing process is initiated when an agent notifies the SIP Application Server that he or she is available to receive calls, at which time the predictive dialer initiates the outbound call.

The following sample call flow is based on call analysis being implemented in the Contact Center. Note that the scenario is the same whether SIP trunking is involved, or a VoIP gateway using T1 trunking.

An explanation follows the diagram.

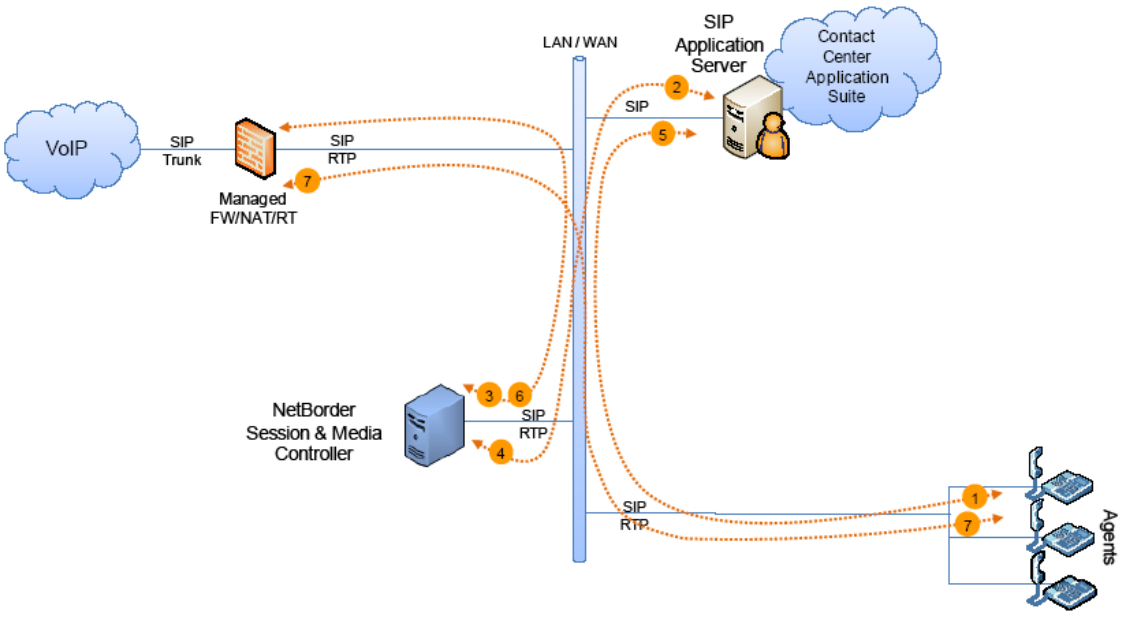


The steps involved in the scenario depicted are as follows:

1. Agents register (using SIP REGISTER) to notify the SIP Application Server of their availability to receive calls.
2. Predictive Dialer initiates the outbound call via the SIP Application Server.
3. NetBorder receives the request and re-originates the outbound call towards the VoIP provider network. Note that as soon as media flows (early media), the NetBorder Netborder Call Analyzer starts processing call information.
4. NetBorder passes NCA results to the SIP Application Server. Based on the NCA results (Voice), the SIP Application Server looks for the next available agent.
5. The SIP Application Server invites the selected agent.
6. After proper acknowledgements, NetBorder re-invites the VoIP provider's network to point the session towards the selected agent.
7. Media flows between agent and customer.

Throughout the call, NetBorder stays in line with the SIP call control path, but gets out of the loop on the audio stream as soon as call progress is complete.

Another way to depict the same scenario is provided below (a network incorporating TDM traffic and a VoIP gateway would be the same).



Chapter 5: Configuring the application

This chapter describes how to configure the NetBorder Call Analyzer, and how to monitor its performance and results using log files.

This chapter contains the following topics:

Selecting a mode of operation

There are three modes of operation recommended when using the NetBorder Call Analyzer :

- **Initial mode:** “Out-of-the-box”, the application is configured to give you the required flexibility and freedom for development purposes.
- **Data collection mode:** Emphasis is placed on gathering high-quality, reliable and pertinent data.
- **Production mode:** Many of the options associated with the previous two modes of operation are disabled, and the application is configured for optimum performance.

Call progress analysis is carried out during each mode of operation, as required. For example, in data collection mode, you will want to collect NCA results on a wide array of calls to measure, troubleshoot and improve detection performance. Finally, in production mode, you will want to carry out CPA as efficiently as possible for large volumes of calls. For information on how to make a CPA-requested call, see Making a call with CPA on page 41.

Below you will find some guidelines for the three modes of operation. Note that each mode is user-definable: the following are recommendations only.

Initial mode

“Out-of-the-box” the NetBorder Call Analyzer is configured as follows:

- Used as an outbound proxy
- Call logging is enabled.
- Call Recording is not used.

Data collection mode

Typically, data collection will be done in collaboration with NetBorder and can be used to measure, troubleshoot and improve the detection performance of a specific deployment. Specifically, in this mode you should:

- Ensure call logging is enabled (see Enabling/disabling call logs on page 71).

- Activate the following logger, and raise its logging level to INFO, so that CPA data collected may be simulated offline: *log4cplus.logger.netborder.cpa=INFO*

  For procedures, see Enabling logging parameters on page 70.
- Enable Call Recording (see Recording media on page 65).

Production mode

Production mode is the mode to use for deployment. To go into production mode, you should:

- Disable call logging (see Enabling/disabling call logs on page 71).
- If your application no longer requires call recording, disable it (see Recording media on page 65).
- Set the *netborder.sip.message* logger to WARN level (*log4cplus.logger.netborder.sip.message=WARN*) in files *call-analyzer-engine-logging.properties* and *call-analyzer-logging.properties.*

Selecting the Application Class

An Application Class is a pre-packaged set of call scenarios and related parameters and is used to quickly integrate NetBorder Call Analyzer with third-party components. The available Application Classes are:

Name	*Use*
CallAnalyzerAsOutboundProxy	SIP Dialer can set an outbound proxy. See Configuring NetBorder as an outbound proxy below
CallAnalyzerGenesysOCS	SIP Dialer is Genesys Outbound Server (OCS) or cannot set an outbound proxy. See Configuring NetBorder to be used with Genesys SIP Server (ie relay server) on page 60

The Application Class is selected via the *netborder.sip.ua.python.appClass* of the *[NETBORDER_INSTALLDIR]\config\ call-analyzer-service.properties* file. Restart the NetBorder Call Analyzer service to activate changes.

Application Classes comes with a set of associated call parameters. **The parameters used for a given Application Class are read for the file named <*Application Class*> .call-properties.**

For example, call properties for the *CallAnalyzerAsOutboundProxy* class are read from the *config*\ *CallAnalyzerAsOutboundProxy.call-properties* file.

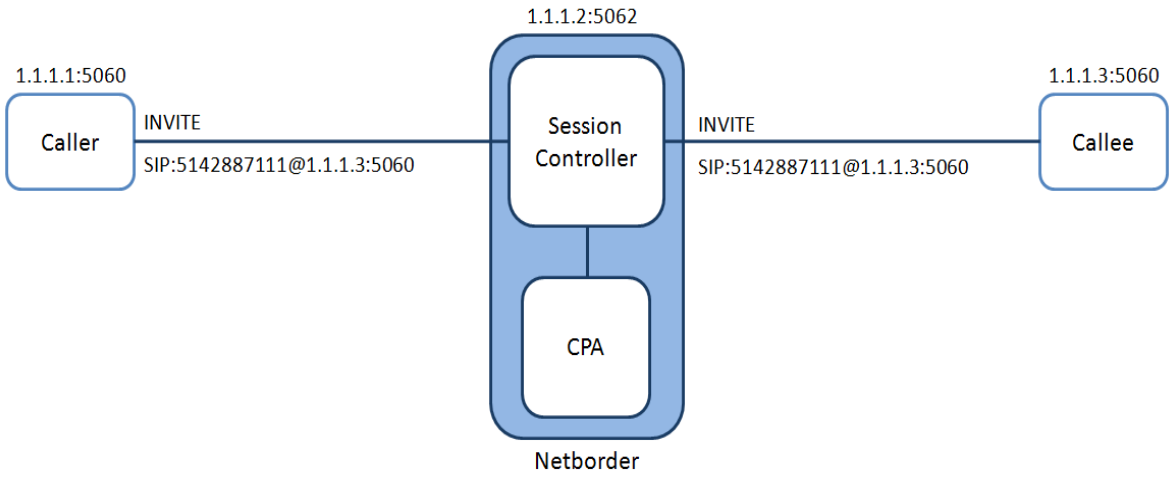
Configuring NetBorder as an outbound proxy

By default, or “out of the box”, the NetBorder Call Analyzer is configured to serve as an outbound proxy.

RFC 3261 defines an outbound proxy as "A proxy that receives requests from a client, even though it may not be the server resolved by the Request-URI. Typically, a UA is manually configured with an outbound proxy".

For information on manually configuring a user agent client with outbound proxy, see Configuring as an outbound proxy on page 35 .

All outbound requests from the UAC or caller pass through the outbound proxy server. The proxy server evaluates them, and if allowed, re-establishes the requests on the outbound side to the UAS or callee. Likewise, responses or initial requests coming from the UAS go to the proxy server to be evaluated. The proxy then communicates with the UAC. In other words, all SIP requests and responses are sent through the outbound proxy; in this case, NetBorder.

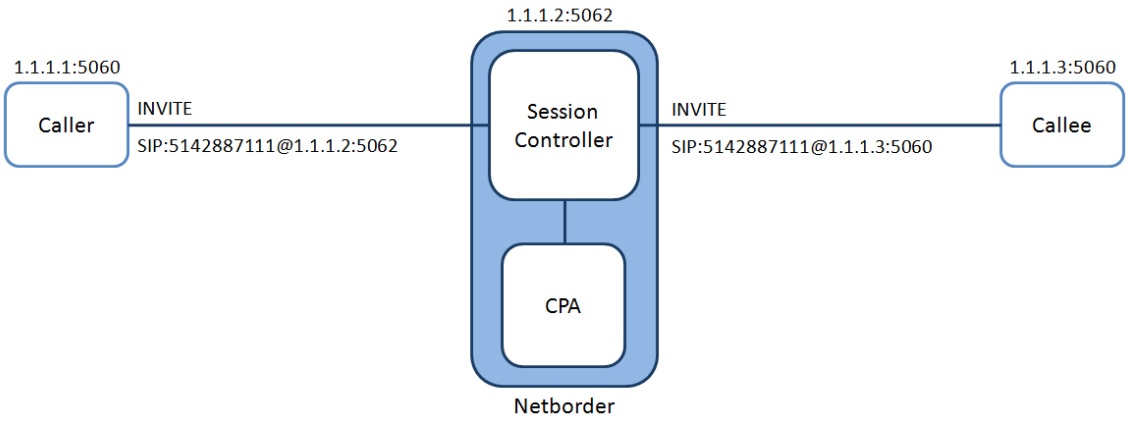


Configuring NetBorder to be used with Genesys SIP Server (ie relay server)

Since Genesys SIP Server does not support setting an outbound proxy, NetBorder Call Analyzer can be configured in a special mode where it is used as a relay server to overcome that limitation. Other SIP-based dialers that does not support settings an outbound proxy can use that mode.

When NetBorder Call Analyzer is configured to use a relay server, incoming requests are mapped to a pre-determined IP address and port at real-time. For every initial SIP request, the user part and parameters of the Request-URI header are copied from the initial SIP request and sent on to the relay server; for example, to the IP address and port of a gateway.

Also, calls that are not meant to be analyzed (ie without ;cpd=on), will be redirected via a 302 SIP response to avoid taking resources on NetBorder Call Analyzer.



To configure NetBorder Call Analyzer for Genesys SIP Server (i.e. as a relay server):

- Using the text editor of your choice, open the file *call-analyzer-service.properties* located at:
  - *[NETBORDER_INSTALLDIR]\config\ call-analyzer-service.properties*

    where *[NETBORDER_INSTALLDIR]* is the root folder of the installation (for example, *C:\Program Files\Sangoma NetBorder Platform 2.0*) .
  - Set the parameter *netborder.sip.ua.python.appClass* to *cpa.CallAnalyzerGenesysOCS*

- Using the text editor of your choice, open the file *CallAnalyzerGenesysOCS.call-properties* located at:
  - *[NETBORDER_INSTALLDIR]\config\ CallAnalyzerGenesysOCS.call-properties* .

  For more information on this file, see Appendix B .

Set the appropriate values for the host and port values, save your changes and close the file.

1. When your changes are complete, restart the NetBorder Call Analyzer service.

To use a relay server, the following parameters must be enabled:

- app.nca.RelayServerHost
- app.nca.RelayServerPort

These parameters must be configured in the *.call-properties* file. (ex: *CallAnalyzerGenesysOCS.call-properties*).

The pertinent default content of the *CallAnalyzerGenesysOCS.call-properties* file is provided below.

```
# Call parameters for Call Analyzer use with Genesys Outbound Contact Server (OCS)

# Host:Port of SIP UAS (media gateway or SIP trunk) used to reach the PSTN
#
# **** Host must be set by entering host name or IP after the "=" sign ***
#      Ex. app.nca.RelayServerHost=192.168.1.100
#
app.nca.RelayServerHost=<Replace by relay server hostname or IP>
app.nca.RelayServerPort=5060

# Enable End Of Greeting (aka EAMD) detection
# Default: "false"
app.nca.EndOfGreeting.enable=false

# Require ";cpd=on" parameter in the initial INVITE request URL to perform
#   Call Analysis (CPA)
# If set to "false", all calls will have CPA performed
# Default: "true"
app.nca.RequireCpdOnToPerformCpa=true

# Accept calls even when Call Analysis is not available
# Default: "false"
app.nca.AcceptCallsWhenCPANotAvailable=false
```

```
# Timeouts

app.nca.PreConnectTimeout.ms=26000

# Used only when End Of Greeting detection is enabled
app.nca.EndOfGreetingTimeout.ms=60000

# Thresholds
#
app.nca.HumanThreshold=0.75
app.nca.MachineThreshold=0.85
app.nca.FaxThreshold=0.85

# UDP host:port to reach the Call Analyzer Engine service
app.nca.EngineHost=127.0.0.1
app.nca.EnginePort=5063
```

Configuring End Of Greeting Detection (EAMD)

When using End Of Greeting Detection mode, if the greeting of an answering machine is detected, the 200 Ok SIP response with the *CPD-Result: Answering Machine* header will be sent only once the end of the greeting is detected (whether via beep sound, or a period of silence), therefore allowing voice message broadcasting campaigns.

There is a timeout parameter provided to set the maximum waiting time allowd before the end of greeting to is detected. If this timeout expired, the 200 Ok SIP response with the *CPD-Result: Answering Machine* header will be returned and more information included in the *X-Netborder-Detailed-CPD-Result* header. See sections Timeout Sequence and X-Netborder-Detailed-CPD-Result for addition informations.

To use the End Of Greeting mode of the *CallAnalyzerAsOutboundProxy* Application Class:

- Using the text editor of your choice, open the call properties in the *[NETBORDER_INSTALLDIR]\config\ CallAnalyzerAsOutboundProxy.call-properties files*
- Set *app.nca.EndOfGreeting.enable* to *true*
- Restart the *Sangoma NetBorder Call Analyzer* service

To use the End Of Greeting mode of the *CallAnalyzerGenesysOCS* Application Class:

- Using the text editor of your choice, open the call properties in the *[NETBORDER_INSTALLDIR]\config\ CallAnalyzerGenesysOCS.call-properties files*
- Set *app.nca.EndOfGreeting.enable* to *true*
- Restart the *Sangoma NetBorder Call Analyzer* service

Configuring the country used for tone definitions

The tones using in telephony signaling differs from country to country. The detection engine underneath NetBorder Call Analyzer can be configured to handle tones for a large number of countries.

The default is to used tone definitions for Canada, which are also good for the U.S.A. If calls are to be placed outside of those two countries, you need to set the country tone definitions to use.

This is done via the *netborder.cpa.runtime.model.country* property of the NetBorder Call Analyzer engine. See how to set engine parameters in section call-analyzer-engine.properties on page 84

Changing the SIP transport to TCP

The default SIP transport use by NetBorder Call Analyzer is UDP. To change the transport to TCP:

- Change **udp** for **tcp** in the netborder.sip.userAgent.IPAddress parameter value in files:
  - *[NETBORDER_INSTALLDIR]\config\ call-analyzer-service.properties*
  - *[NETBORDER_INSTALLDIR]\config\ call-analyzer-engine.properties*
- *Restart the NetBorder Call Analyzer service.*

Returning NetBorder to its original configuration

If you change the configuration files—for example, to use a relay server—and find that you need to return them to their default settings, the original configuration files are available in the following folder:

- *[NETBORDER_INSTALLDIR]\.original_config*

  where *[NETBORDER_INSTALLDIR]* is the root folder of the installation (for example, *C:\Program Files\Sangoma NetBorder Platform 2.0\.original_config*).

You can either copy the pertinent files to their related folders (such as the *\config* folder for *.properties* files, .call-properties and license files), provided you have made no other significant changes, or you can use the original configuration as a reference point.

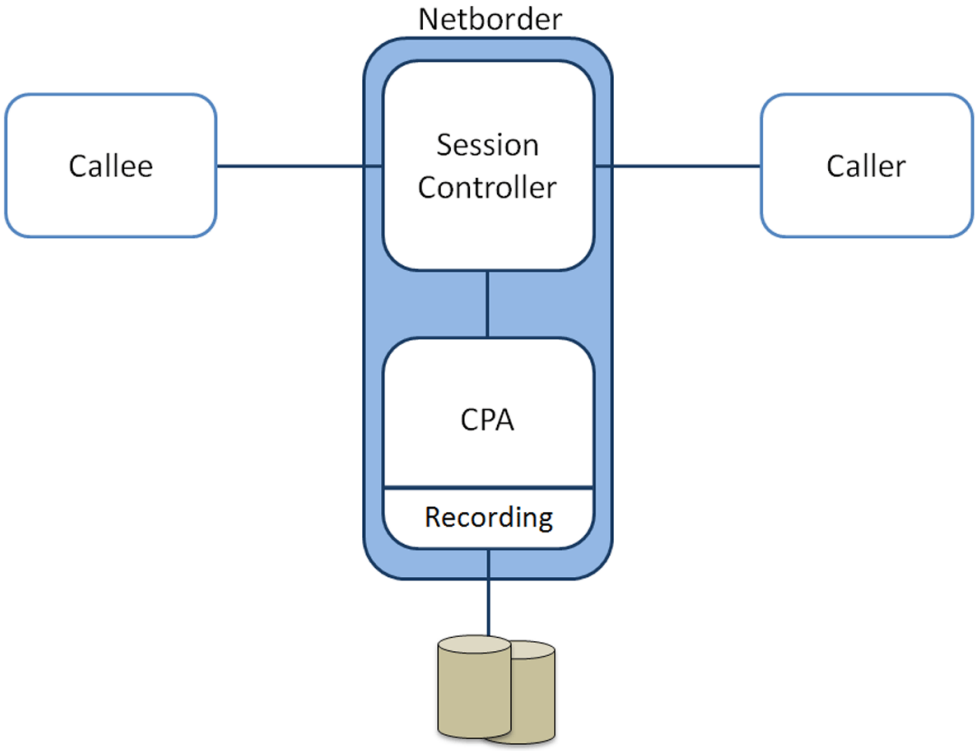
Recording media

The NetBorder Call Analyzer allows you to record media automatically. You can do so by enabling the:

- **CPA embedded recorder:** Allows you to record media until call progress analysis is complete.

Using the CPA embedded recorder

To use the embedded recorder on CPA-requested calls, enable the *netborder .cpa.runtime.recordAudio* parameter in the *call-analyzer-engine.properties* file.



To enable the CPA embedded recorder:

1. Using the text editor of your choice, open the file *call-analyzer-engine.properties* file located at:
   - *[NETBORDER_INSTALLDIR]\config\ call-analyzer-engine.properties*

     where [NETBORDER_INSTALLDIR] is the root folder of the installation (for example, C:\Program Files\Sangoma NetBorder Platform 2.0\config\ call-analyzer-engine.properties).

   For more information on this file, see Appendix B .
1. Search for the following text string:
   - *netborder.cpa.runtime.recordAudio*
1. To enable the *netborder.cpa.runtime.recordAudio* parameter, change the value of the parameter to "*true*". For example:
   - *netborder.cpa.runtime.recordAudio=true*
1. Save your changes and close the file.
2. Restart the NetBorder Call Analyzer service.

Setting up logging

The NetBorder Call Analyzer includes a powerful logging framework to enable you to control the logging of events.

Logging configuration is set in the following two main logging configuration files:

- *[NETBORDER_INSTALLDIR]\config\call-analyzer-logging.properties* (for the Sangoma NetBorder Call Analyzer service)
- *[NETBORDER_INSTALLDIR]\config\call-analyzer-engine-logging.properties* (for the Sangoma NetBorder Engine for Call Analyzer service)

  where [NETBORDER_INSTALLDIR] is the root folder of the installation.

Separate files allow you to maintain two custom logger configurations, one for the Call Analyzer and one for the Engine for Call Analyzer service.

NOTE: Detailed information on logging and particularly logging configuration is contained in Appendix C. Turn to this appendix to familiarize yourself with logging levels, the logging subsystem, and configuration.

Viewing logs and events

This section describes briefly how to view logging information.

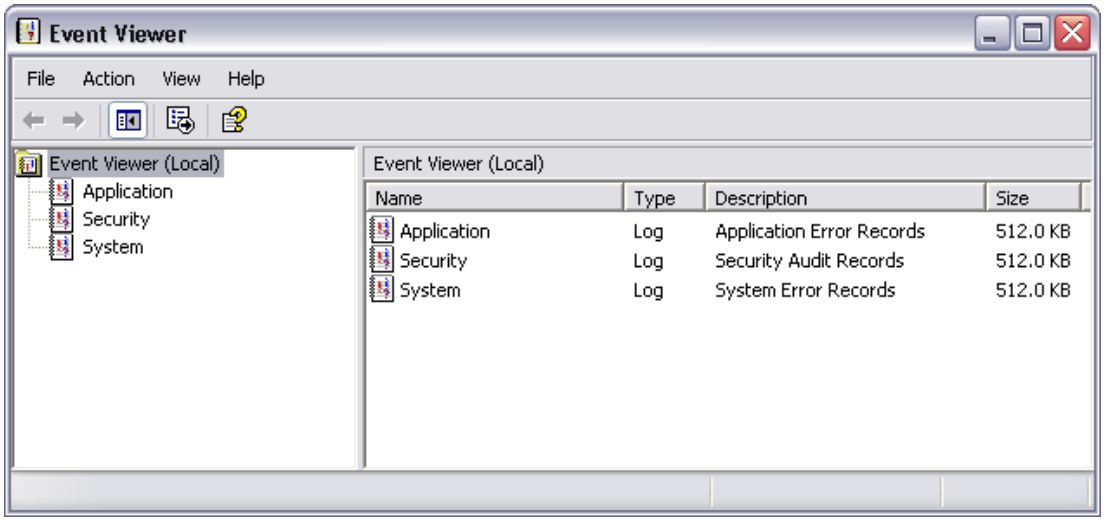
Viewing logs of high-level events

By default, and unless otherwise changed in the logging configuration, all logs of severity “WARN” and above are reported immediately, through the Windows Event Viewer.

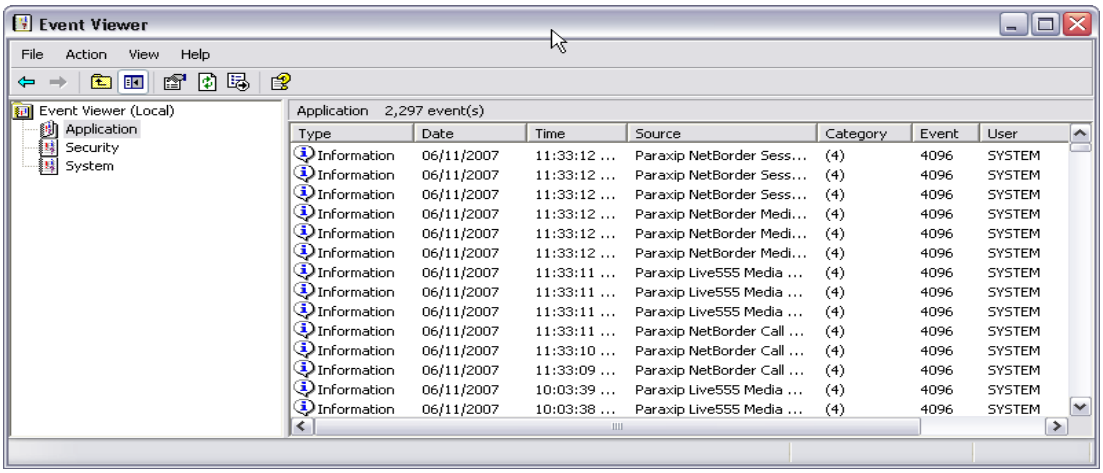
To monitor the service to ensure that no warnings or errors occur, start the Event Viewer.

To use the Windows Event Viewer to view logging information:

1. From the **Start Menu**, select **Programs** > **Sangoma NetBorder Platform 2.0** > **Event Viewer**.
2. In the left pane, select the “Application” folder.



3. In the right pane, click the “Source” event header to view events in ascending (alphabetical) order.



4. Search through the list for information and/or any errors resulting from a “Sangoma NetBorder” service.

Viewing logs of low-level events

By default, logs of lesser priority (such as “INFO”) are reported to a file, as configured in the *.properties* files.

For example, call logs are configured as follows:

```
# Call Logging
log4cplus.appender.CALL_LOG_APPENDER=log4cplus::NullAppender
log4cplus.appender.CALL_LOG_APPENDER.Directory=logs/call-logs
log4cplus.appender.CALL_LOG_APPENDER.ImmediateFlush=true
log4cplus.appender.CALL_LOG_APPENDER.layout=log4cplus::Pattern
Layout
log4cplus.appender.CALL_LOG_APPENDER.layout.ConversionPattern=%D{%Y-%m-
%d %H:%M:%S:%%q} [%t] %p - %c : %m%n
# Set to true if you want a directory structure with
# year/month/day/hour for your call logs
netborder.infra.CallLogger.dateTimeDirectory=true
# Overload ROOT with new logger hierarchy severity
log4cplus.logger.netborder.sip.message=INFO
log4cplus.logger.netborder.ace=WARN
```

In this example, to view SIP message events at the “INFO” level, you would look for log files in subdirectories organized by date and time in the following parent directory:

- *C:\Program Files\Sangoma NetBorder Platform 2.0\logs\call-logs*

By default *Engine For Call Analyzer* events of level “INFO” are saved to the *[NETBORDER_INSTALLDIR]\logs\call-analyzer-engine.out* log file, as specified in the *call-analyzer-engine-logging.properties* file. Similarly, for the *Call Analyzer*, logging information of level “INFO” is saved to the *[NETBORDER_INSTALLDIR]\logs\call-analyzer-service.out* log file, as per the *call-analyzer-logging.properties* file.

However, logs can be redirected to any file you wish, simply by modifying the *formatting handle*, also called an “*appender*”, which holds the information on where to redirect the logging output (for example, Windows Event Viewer, console, file, *syslog*), as well as the type and format of logging information to output. For more information, see Appendix C and specifically Step 1: Set the logging level and appender on page 93.

Enabling logging parameters

The *.properties* files contain logging parameters that may be enabled, as necessary. In most cases, you will change these parameters only when instructed to do so by Sangoma Support.

Additional Logging Parameters	
Parameter	***Description***
netborder.sip.message	Enables logging related to SIP messages exchanged.
netborder.sip-hub	Enables logging related to the core of NetBorder.
netborder.cpa	Enables logging for CPA decision-related events.

Follow the steps of the following procedure whenever you need to enable (or disable) a logging parameter in one of the *.properties* files (altering the appropriate parameter as required).

This procedure describes how to enable the *netborder.cpa* logger in the *sip-hub.properties* file and change its logging severity level from “TRACE” to “INFO”.

To enable the netborder.*cpa* logger:

1. Using the text editor of your choice, open the *call-analyzer-logging.properties* file, located at:
   - *[NETBORDER_INSTALLDIR]\config\ call-analyzer-logging.properties*

     where *[NETBORDER_INSTALLDIR]* is the root folder of the installation (for example, *C:\Program Files\Sangoma NetBorder Platform 2.0\config\ call-analyzer-logging.properties).*
2. Search for the following text string:
   - *log4cplus.logger.netborder.cpa*
1. To enable the property, remove the character “#” from the beginning of the line. For example:
   - *log4cplus.logger.netborder.cpa=TRACE*

   (Therefore, to disable a logging parameter, precede it with the ‘#’ character, so it is ignored by the system.)
2. Change the logging level. In the example below, the logging level has been changed from “TRACE” to “INFO”:
   - *log4cplus.logger.netborder.cpa=INFO*

CAUTION: The performance of the system is significantly degraded when the log level is set to TRACE. DEBUG and TRACE should only be used by Sangoma Support when troubleshooting a specific issue.

1. Save your changes and close the file.
2. When your changes are complete, restart the NetBorder Call Analyzer service.

Enabling/disabling call logs

By default call logging is enabled.

To disable call logging:

1. With the text editor of your choice, open each of the following configuration files:
   - *[NETBORDER_INSTALLDIR]\config\ call-analyzer-logging.properties* (for the Sangoma NetBorder Call Analyzer service)
   - *[NETBORDER_INSTALLDIR]\config\ call-analyzer-engine-logging.properties (for the Sangoma NetBorder Engine For Call Analyzer service)*
   - *[PARAXIP_INSTALLDIR]\config\media_server_uas.properties (for the Paraxip NetBorder Media Server service)*

     where *[NETBORDER_INSTALLDIR]* is the root folder of the installation (for example, *C:\Program Files\Sangoma NetBorder Platform 2.0\config\ call-analyzer-logging.properties*) .
1. Search for the text string "*log4cplus.rootLogger*" to locate the root logger:
   - *log4cplus.rootLogger=WARN, NTEVENTLOG, DEBUG_APPENDER, CALL_LOG_APPENDER*
1. Comment out the ", CALL_LOG_APPENDER" value by placing the character "#" directly in front of it, on a new line, as follows:
   - *log4cplus.rootLogger=WARN, NTEVENTLOG, DEBUG_APPENDER #, CALL_LOG_APPENDER*
1. Make this change to each of the *.properties* files.
2. Save your changes and close the files.
3. Restart the NetBorder Call Analyzer service.

To enable the call log once again, ensure that “, CALL_LOG_APPENDER” is not commented out. Restart the service.

Chapter 6: Troubleshooting

List of errors messages and associated actions to remedy to the problem:

Component	Error Message	Severity	Action
Call Analyzer	Cannot parse NCA result = %s	High	Internal error, contact *Sangoma* Support
NetBorder	Error in Python script: %s	High	Correct Python code following reported error details
NetBorder	Error instantiating Python class	High	Correct Python code for specified class following reported error details
NetBorder	Exception while configuring Resiprocate:	High	Correct parameters following reported exception text
Infrastructure	Failed to configure the logger from the Global Configuration	High	Correct logger-related parameters in /config/[service].properties
Infrastructure	Host has 2 IP addresses.	High	Set the netborder.net.primaryIPAddress parameter
Infrastructure	Host has more than one IP address	High	Set the netborder.net.primaryIPAddress parameter
Call Analyzer	No valid NCA result in = %s	High	Internal error, contact *Sangoma* Support
Call Analyzer	doReInvite on callee failed	High	Check SIP messaging to see why the Callee User agent refused the re-INVITE once connected
Infrastructure	failed to create call logging output directory	High	Check permissions for path provided to store call logs
Infrastructure	Failed to load config files to initialize	High	Make sure configuration file provided by error message exists and can be opened

Infrastructure	host has no ip address!	High	Make sure at least only network adatper is bound to the TCP/IP protocol
Infrastructure	The Global Configuration contains deprecated parameters. Please update your configuration	High	Remove deprecated parameters (provided by WARN-level logs)
Infrastructure	License is expired. Expiration date: %s	High	Contact *Sangoma* Sales to obtain a permanent license
Infrastructure	License allows the software to be run on a host with an Network adapter using a MAC address of only. MAC addresses for this host:%s	High	Contact *Sangoma* Sales to obtain a valid license for your server
Infrastructure	Signature validation on license file %s failed	High	Make sure you update both the license file and it's signature
Call Analyzer	CPA endpoint disconnected unexpectedly	Medium	Internal error, contact *Sangoma* Support
Call Analyzer	The SIP UAC did not answer our SIP 200 response with a ACK request as specified by RFC 3261. Terminating the call	Medium	Check configuration of Caller SIP User Agent

Appendix A: Glossary

This appendix contains a list of terms, abbreviations and acronyms used in this guide. Definitions of key terms have been provided, many of which have been taken directly from the SIP standard (RFC 3261).

A

ACD	Automatic Call Distribution/Distributor. ACD efficiently routes incoming calls to multiple answering stations. ACDs also enable call centers to track usage patterns as well as traffic and agent performance.

B

B2BUA	Back-to-Back User Agent. A B2BUA acts as third-party call controller and can establish calls between two user agents, as well as modify and terminate existing calls. SIP calls via a B2BUA result in the creation of two distinct dialogs, which allow it to modify one session without affecting the other session. A B2BUA is a user agent (*UA*) that acts as a user agent server (*UAS*) to the *Caller* , and as a user agent client (*UAC*) to the *Callee* .

C

Call	A call is an informal term that refers to some communication between peers, generally set up for the purposes of a multimedia conversation.
Callee	The callee is the party that receives an INVITE request for the purpose of establishing a new session. In SIP terms, a *UAS* .
Caller	The caller is the party initiating a session (and dialog) with an INVITE request. In SIP terms, a *UAC .*
Call leg	See *Dialog* .
Client	A client is any network element that sends SIP requests and receives SIP responses. Clients may or may not interact directly with a human user. UACs and proxies are clients.
Contact Center	A Contact Center or call center is a centralized office used for the purpose of receiving and transmitting a large volume of requests by telephone.
CPA	Call Progress Analysis. CPA is the process of detecting pre-connect information about failed outbound call attempts and the destination party's media type for connected outbound calls.

D

Dialer	A dialer, also referred to as a *Predictive Dialer*, is a computerized system that automatically dials batches of telephone numbers; for instance, for connection to agents assigned to sales or other campaigns.
Dialog	A dialog is a peer-to-peer SIP relationship between two UAs that persists for some time. A dialog is established by SIP messages, such as a 2xx response to an INVITE request. *Call leg* is another term for a dialog.

E

Early media	Early media is the ability of two user agents to communicate before a SIP call is actually established. Specifically, it permits the delivery of a media stream prior to call answer or session establishment.

H

Header	A header is a component of a SIP message that conveys information about the message. It is structured as a sequence of header fields.

I

IETF	Internet Engineering Task Force. The IETF is an open standards organization that develops and promotes Internet standards, dealing in particular with standards of the TCP/IP and Internet Protocol suite.
IP	Internet Protocol. IP is a protocol used for communicating data across packet-switched networks.
IVR	Interactive Voice Response. IVR is an automated telephony system that interacts with callers, gathers information, and routes calls to the appropriate recipient.

L

LAN	Local Area Network. A LAN is a computer network covering a small geographic area, such as a home, office, or group of buildings.

M

MAC address	Media Access Control address of a computer networking device.
Message	Data sent between SIP elements as part of the protocol. SIP messages are either requests or responses.
Method	The method is the primary function that a request is meant to invoke on a server. The method is carried in the request message itself. Example methods are INVITE, REFER and BYE.
ms	Millisecond. A millisecond (ms) is one thousandth of a second.

O

OAM	Operations, Administration, and Maintenance. OAM is a general term used to describe the processes, activities, tools, standards, etc. involved with operating, administering, and maintaining any system.
OS	Operating System. The software that manages the hardware and software of a computer.
Outbound proxy	A proxy that receives outbound requests from a client, even though it may not be the server resolved by the Request-URI. Typically, a UA is manually configured with an outbound proxy.

P

PBX	Private Branch eXchange. A PBX is a telephone exchange that serves a particular business or office, as opposed to one that a common carrier or telephone company operates for many businesses or for the general public.
Ping	Ping is the primary TCP/IP command used to troubleshoot connectivity, reachability, and name resolution.
Predictive dialer	See *Dialer* .

Proxy	A proxy is an intermediate entity in the SIP network that is responsible for forwarding SIP requests to the target UAS or another proxy on behalf of the UAC. A proxy primarily provides the routing function in the SIP network.
PSTN	Public Switched Telephone Network. The PSTN is the network of the world's public circuit-switched telephone networks.
Python	Python is a dynamic object-oriented programming language distributed under an OSI-approved open source license.

R

Redirect server	A redirect server is a UAS that generates 300-class SIP Responses to requests it receives, directing the UAC to contact an alternate set of URIs.
Registrar server	A registrar is a UAS that accepts SIP REGISTER requests and updates the information from the request message into a location database.
Relay server	Relay servers facilitate communication by acting as proxies for navigating firewalls, providing alternative communications paths for clients operating over gateways, and fanning out data transmissions.
Request	A SIP message sent from a client to a server, for the purpose of invoking a particular operation.
Request-Line	The Request-Line is the start-line of a SIP request. If CPA has been enabled using the NetBorder Call Analyzer, by default the Request-Line will contain the string "`cpd=on`".
Response	A SIP message sent from a server to a client, for indicating the status of a request sent from the client to the server.
RFC	Request for Comments associated with an active IETF Working Group.
RTCP	RTP Control Protocol. RTCP permits monitoring of the data delivery of RTP, including quality of service.
RTP	Real-time Transport Protocol. RTP transports real-time data such as audio or video packets to the endpoints that are involved in a session. RFC 3550 defines RTP.

S

SBC	See *Session Controller .*
SDP	Session Description Protocol. SDP is used in a SIP message body to describe the parameters of the multimedia session. RFC 2327 defines SDP.
Server	A server is a network element that receives requests in order to service them and sends back responses to those requests. Examples of servers are proxies, user agent servers, redirect servers, and registrars.

Session	A multimedia session is a set of multimedia senders and receivers and the data streams flowing from senders to receivers. A multimedia conference is an example of a multimedia session.
Session Controller	A Session Controller or *Session Border Controller* (SBC) is a piece of network equipment or a collection of functions that control real-time session traffic at the signaling, call control, and packet layers as they cross a notional packet-to-packet network border between networks or between network segments.
SIP	Session Initiation Protocol. SIP is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. RFC 3261 defines SIP.
SIT	Special Information Tone. SIT is a series of three precise, sequential audio tones played together, each tone having a higher pitch than the previous tone, which indicate that a call cannot be completed.
Status-Line	The Status-Line is the start-line of a SIP response.

T

TCP	Transmission Control Protocol. TCP is intended for use as a highly reliable host-to-host protocol between hosts in packet-switched computer communication networks, and especially in interconnected systems of such networks.
TDM	Time Division Multiplexing. TDM is a method of sending multiple digital signals along a single telecommunications transmission path.

U

UA	User Agent. A UA is a logical function in the SIP network that initiates or responds to SIP transactions. A UA can act as either the client or the server in a SIP transaction.
UAC	User Agent Client or *Caller* . A UAC is a logical function that initiates SIP requests and accepts SIP responses.
UAS	User Agent Server or *Callee* . A UAS is a logical function that accepts SIP requests and sends back SIP responses.
UDP	User Datagram Protocol. Using UDP, programs on networked computers can send short messages sometimes known as datagrams (using Datagram Sockets) to one another. UDP is also referred to as the *Universal Datagram Protocol*.
URI	Uniform Resource Identifier. A URI is a compact string of characters used to identify or name a resource.

URL	Uniform Resource Locator. Frequently, a synonym for URI. Strictly speaking, a URL is a URI that, in addition to identifying a resource, provides means of acting upon or obtaining a representation of the resource by describing its primary access mechanism or network "location".

V

VAD	Voice Activity Detection. VAD is the process of separating conversational speech and silence. In VoIP, VAD can disable the silence packets and use the silent period to transmit some traffic other than voice.
VoIP	Voice over IP. VoIP is the routing of voice conversations over the Internet or through any IP-based network.

W

WAN	Wide Area Network. A network that uses routers and public communications links. The largest and most well-known example of a WAN is the Internet.

X

XML	eXtensible Markup Language. The Extensible Markup Language is a general-purpose markup language classified as an extensible language because it allows its users to define their own elements or tags. Its primary purpose is to facilitate the sharing of structured data across different information systems.

Appendix B: Configuration parameters

This appendix contains a comprehensive list of parameters, with a brief description, for the following configuration files:

call-analyzer-service.properties

The following table lists the configuration parameters contained in the call-analyzer-service.*properties* file, located at:

- *[NETBORDER_INSTALLDIR]\config\call-analyzer-service.properties*

  where *[NETBORDER_INSTALLDIR]* is the root folder of the installation (for example, *C:\Program Files\Sangoma NetBorder Platform 2.0\config\call-analyzer-service.properties).*

The call-analyzer-service.*properties* file is a simple text file that you can edit using any standard text editor.

Note that the service must be re-started for modifications to take effect. The default values should be suitable for most applications.

NOTE: Lines starting with the character '#' are comments ignored by the software.

Configuration Parameters: call-analyzer-service.properties		
Parameter	***Default Value***	***Notes***
netborder.sip.userAgent.IP Address	INADDR_ANY:5062/udp	Specifies the IP address, port and type of transport used for SIP messages. Format : ip_addr1[:port1][/[udp|tcp]], ip_addr2[:port2][/[udp|tcp]] For example: 127.0.0.1:5062/udp. Use *INADDR_ANY* as the IP address to listen on all IP interfaces on the host. **Note:** Value must be different from the other Sangoma services running on the same host.

Configuration Parameters: call-analyzer-service.properties		
Parameter	***Default Value***	***Notes***
netborder.net.primaryIP Address		Specifies the IP address of the media engine. Use this parameter when you have multiple network cards to specify the one to use.
netborder.sip.ua.python. appClass	cpa.CallAnalyzerAsOutboundProxy	Specifies the Application Class to use to handle the call request. See Selecting the Application Class on page 58
netborder.oam.webServicePort	18082	Port used for Web Service OAM interface
netborder.infra.coreDump.write OnCrash	true	Specifies whether a dump file is generated should the program crash. Valid values: true or false.
netborder.infra.coreDump.write OnCrash.path	[NETBORDER_INSTALLDIR]\config\call-analyzer.dmp	Specifies the location of the dump file if the previous parameter is set to “true”.

call-analyzer-engine.properties

The following table lists the configuration parameters contained in the call-analyzer-engine.propertiesfile, located at:

- *[NETBORDER_INSTALLDIR]\config\call-analyzer-engine.properties*

  where *[NETBORDER_INSTALLDIR]* is the root folder of the installation (for example, *C:\Program Files\Sangoma NetBorder Platform 2.0\config\call-analyzer-engine.properties).*

The *call-analyzer-engine.properties* file is a simple text file that you can edit using any standard text editor. Note that many of the parameters in this file are the same as those contained in the c*onfig\call-analyzer-service.properties file*. For more information on these parameters, see the Table entitled Configuration Parameters: *call-analyzer-service* .properties in the preceding section.

Parameters unique to the *call-analyzer-engine.properties* file are described in the following table.

Note that the service must be re-started for modifications to take effect. The default values should be suitable for most applications.

NOTE: Lines starting with the character '#' are comments ignored by the software.

Configuration Parameters: call-analyzer-engine.properties		
Parameter	***Default Value***	***Notes***
netborder.cpa.runtime.model .country	canada	Geographic location for tone detection. See *[NETBORDER_INSTALLDIR]*/data/tone-db for the list of locales supported. NOTE: "canada" and "usa" are equivalent, see Configuring the country used for tone definitions on page 63
netborder.cpa.runtime.numThreads	2	Adjust to the number of logical CPUs on the system

Configuration Parameters: call-analyzer-engine.properties		
Parameter	***Default Value***	***Notes***
netborder.cpa.runtime.sendrecvSDPOffer	FALSE	Make Call Analyzer offer the RTP stream as bidirectional (sendrecv) in the initial INVITE request (default is unidirectional - recvonly)
netborder.media.rtp.udpPortRange	19000:22000	Sets the port range to use for RTP & RTCP transport
netborder.cpa.runtime.recordAudio	false	Specifies whether to record the audio received by the Netborder Call Analyzer. Valid values: true or false
netborder.recorder.outputDirectory	[NETBORDER_INSTALLDIR]\logs\recordings	Specifies the directory used to store the audio files recorded.
netborder.recorder.outputDirectory.dateTime	true	Indicates whether the audio files are saved in directories structured by date and time. Valid values: true or false

call-properties files

This section describes the configuration parameters contained in the *call-properties* files, located at:

- *[NETBORDER_INSTALLDIR]\config*

  where *[NETBORDER_INSTALLDIR]* is the root folder of the installation (for example, *C:\Program Files\Sangoma NetBorder Platform 2.0\config\CallAnalyzerAsOutboundProxy.call-properties)* .

Here are the call-properties files present in the /config folder:

- *CallAnalyzerAsOutboundProxy.call-properties*
- *CallAnalyzerGenesysOCS.call-properties*

The *.call- properties* file are a simple text file that you can edit using any standard text editor.

Note that the service must be re-started for modifications to take effect.

NOTE: Lines starting with the character '#' are comments ignored by the software. Paragraphs or blocks of text are commented out by means of triple quotes; that is, the text to be ignored is surrounded by three sets of quotations marks (""").

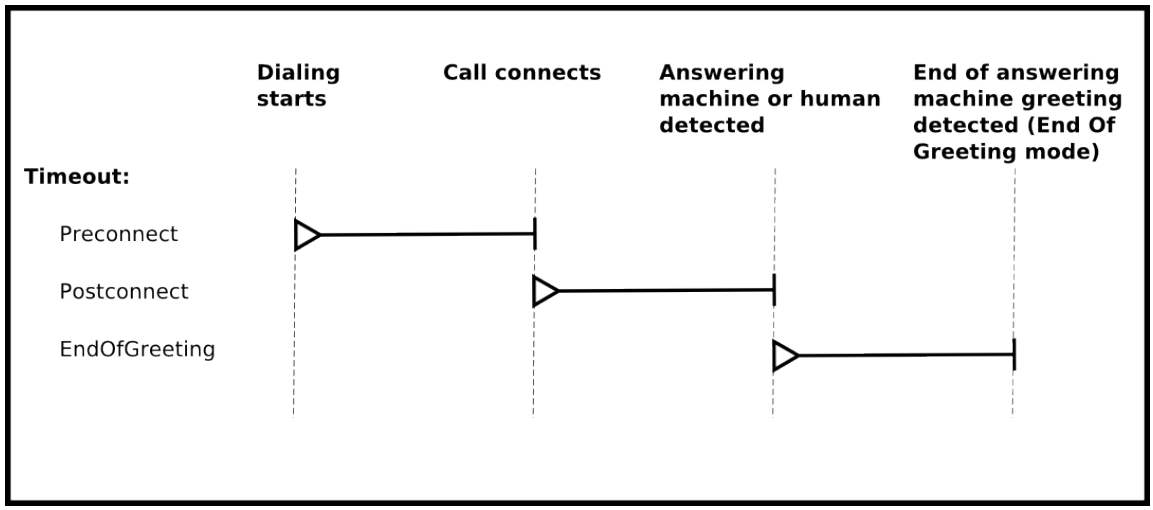
*Configuration Parameters: * .call- properties*		
Parameter	***Default Value***	***Notes***
app.nca.PreConnectTimeout.ms	26000	The maximum amount of time (in milliseconds) before a connect event or a NCA result is obtained. If the timeout is reached, an "Unknown" result is returned to the caller.
app.nca.PostConnectTimeout.ms	7000	The maximum amount of time (in milliseconds) for a NCA result to be obtained, once connected. If the timeout is reached, an "Unknown" result is returned to the caller. The post-connect timeout in no way can enhances detection speed: it is only a fail safe to ensure that the CPA does not stay locked on a call indefinitely. If the End Of Greeting mode is not in use, the maximum total timeout time is equal to: app.nca.PreConnectTimeout.ms + app.nca.PostConnectTimeout.ms.
app.nca.EndOfGreetingTimeout.ms	60000	The maximum amount of time (in milliseconds) for a End of Greeting result to be obtained when the End Of Greeting mode is active. This timeout starts once when Netborder Call Analyzer detects an answering machine). Refer to the Timeout Sequence Graph to get an example of a timeout sequence. When using the End Of Greeting detector, the maximum total timeout time is equal to: app.nca.PreConnectTimeout.ms + app.nca.PostConnectTimeout.ms + app.nca.EndOfGreetingTimeout.ms.
app.nca.HumanThreshold	0.75	The threshold, specified as a decimal percentage, at which a CPA HUMAN result will be considered a final result and reported to the caller.

*Configuration Parameters: * .call- properties*		
Parameter	***Default Value***	***Notes***
app.nca.MachineThreshold	0.85	The threshold, specified as a decimal percentage, at which a CPA MACHINE result will be considered a final result and reported to the caller.
app.nca.FaxThreshold	0.85	The threshold, specified as a decimal percentage, at which a CPA FAX result will be considered a final result and reported to the caller.
app.nca.EngineHost	127.0.0.1	Specifies the IIP to reach the NetBorder Call Analyzer Engine.
app.nca.EnginePort	5063	Specifies the listening port of the NetBorder Call Analyzer Engine
common.b2bua.outbound.proxy.host	<not set>	Set the outbound proxy host ip to reach callee.
common.b2bua.outbound.proxy.port	<not set>	Set the outbound proxy port to reach callee.
app.nca.RelayServerHost	<not set>	Specifies the IP address of the relay server. For every initial SIP request, the content of the Request-URI header is copied from the initial SIP request and sent on to the relay server. This parameter is used only when the NetBorder Call Analyzer is configured for Genesys SIP Server
app.nca.RelayServerPort	5060	Specifies the listening port of the relay server.
app.nca.RequireCpdOnToPerformCpa	true	Set to false so perform analysis on all incoming calls, I.e do not require cpd=on to be present in Request URL.

Timeout Sequence

This diagram illustrates which timeout applies during which part of the call. Note that in this sequence no timeout is triggered. The horizontal lines represent a timeout that is 'ticking'.

For example, if an answering machine or human were to not be detected before the Postconnect timeout is reached, the call attempt would be stopped with a Detailed-Cpd-Result header value of “Unknown”.



Appendix C: Logging configuration

This appendix contains general information about logging, including the following key topics:

Logging levels

There are six logging levels, as follows:

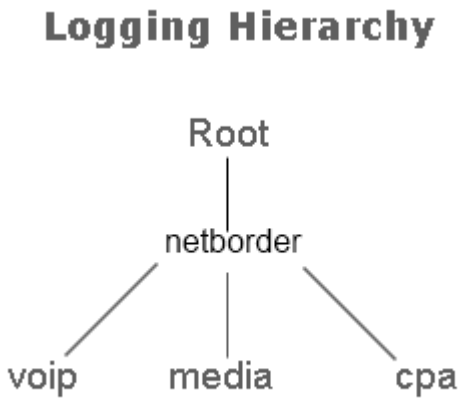
- **FATAL:** Logs very severe error events that may lead the application to abort.
- **ERROR:** Logs only error conditions. The ERROR level provides the smallest amount of logging information.
- **WARN:** Logs information when an operation completes successfully but there are issues with the operation.
- **INFO:** Logs information about workflow. It generally explains how an operation occurs.
- **DEBUG:** Logs all of the details related to a specific operation. This is the highest level of logging.
- **TRACE:** Logs designated finer-grained informational events than DEBUG.

CAUTION: The performance of the system is significantly degraded when the log level is set to TRACE. DEBUG and TRACE should only be used by Technical support when troubleshooting a specific issue.

Logger hierarchy

The logger adheres to a hierarchical structure starting with a “Root” logger. Components that are direct “children” of a logger above them in the hierarchy inherit all of the properties of the parent. The value inherited by a child from the parent can by overridden by modifying the specific property of the child. A change at one level will impact all of the component’s descendants.

The figure below illustrates a subset of the NetBorder Call Analyzer‘s logging hierarchy. Those components that are used solely for troubleshooting purposes by Sangoma Support have not been included (for example, infrastructure components and utilities).



The “public” logging sub-modules of most importance are as follows:

- **VoIP:** Relates to everything that has to do exclusively with signaling within the context of a VoIP dialog.
- **Media:** Relates to everything that has to do exclusively with media processing (for example, media recording).
- **CPA:** Relates to everything that has to do with the general NetBorder Call Analyzer application and is independent of the specifics of the Media and VoIP implementations. It includes, for example, the Call Analyzer logs, Engine Call Analyzer logs, and call logs.

Configuring the logging subsystem

To configure the logging subsystem, follow these steps:

- Step 1: Set the logging level and appender (see page 93)
- Step 2: Set the pattern layout (see page 95)
- Step 3 (optional): Set child-specific behaviour (see page 96).

Step 1: Set the logging level and appender

The first step to configuring the logging subsystem is to set the root logger. The root logger can be assigned a logging level and one or more formatting handles.

A f*ormatting handle*, also called an "*appender"*, holds the information on where to redirect the logging output (for example, Windows Event Viewer, console, file, *syslog*, etc.), as well as the type and format of logging information to output.

Here is a sample configuration for the root logger. Logging parameters are set in the main configuration (.*properties*) files: *call-analyzer-service.properties* and *call-analyzer-engine.properties* .

```
# Logger configuration for the Windows Event log, level = INFO
log4cplus.rootLogger=INFO, NTEVENTLOG, ROLLINGFILE
# NTEVENTLOG Appender
log4cplus.appender.NTEVENTLOG=log4cplus::NTEventLogAppender
# ROLLINGFILE Config: Size limited rolling files of 50MB with 20
backups (max 1GB)
log4cplus.appender.ROLLINGFILE=log4cplus::RollingFileAppender
```

In the example provided, the root logging level is set to INFO, and the appender (formatting handle) is NTEVENTLOG. The appender is configured to redirect the output to the Windows Event Viewer using *NTEventLogAppender* and a *RollingFileAppender*.

To set the target redirection, assign the property *log4cplus.appender.<MY_HANDLE>* (where *<MY_HANDLE>* is the name of the logger's redirection handle) to one or more of the values listed in the following table.

Logging Appenders	
Appender	***Description***
log4cplus::NTEventLog Appender	Output redirected to the Windows Event Viewer.
log4cplus::ConsoleAppender	Output redirected to stdout (standard output).
log4cplus::RollingFileAppender	Output redirected to a rolling file (rolling executed based on the log file size). If this type of appender is chosen, then the following properties can be configured: log4cplus.appender.<MY_HANDLE>.File=<filename>.log log4cplus.appender.<MY_HANDLE>.MaxFileSize=50MB log4cplus.appender.<MY_HANDLE>.MaxBackupIndex=20 log4cplus.appender.<MY_HANDLE>.ImmediateFlush=true
Log4cplus::DailyRollingFile Appender	Output redirected to a scheduled rolling file (rolling executed based on a schedule – hourly, daily, etc.). **There is no restriction on disk space for this appender, so use with care.** If this type of appender is chosen, then the following properties can be configured: log4cplus.appender.<MY_HANDLE>.File=<filename>.log log4cplus.appender.<MY_HANDLE>.Schedule=HOURLY The Schedule property can take one of the following values: MONTHLY, WEEKLY, DAILY, TWICE_DAILY, HOURLY, MINUTELY.
log4cplus::FileAppender	Output redirected to a file (file is overridden each time the service starts). If this type of appender is chosen, then the following properties can be configured: log4cplus.appender.<MY_HANDLE>.File=<filename>.log log4cplus.appender.<MY_HANDLE>.MaxFileSize=50MB log4cplus.appender.<MY_HANDLE>.ImmediateFlush=true

Step 2: Set the pattern layout

After setting the redirection output, the next step is to configure the type and format of the information to appear in the logs. This is achieved through the configuration of a *Pattern Layout***.** A pattern layout allows you to format the output of the logs in a similar fashion to a *printf* function in 'C'. The format string contains one or more placeholders, which will be replaced by the logging engine when it is time to log the message.

Here is an example of how a pattern layout would be assigned to a logger and configured.

```
log4cplus.appender.NTEVENTLOG.layout=log4cplus::PatternLayout
# Output 'Log Level' - 'Logger name' : Message
log4cplus.appender.NTEVENTLOG.layout.ConversionPattern=%p - %c : %m%n
```

In the example above, "%p" indicates the logging level or priority (such as "INFO"), "%c" indicates the source of the event (such as a configuration parameter), followed by the log message itself, and finally a new line.

The following table lists the special conversion characters available for use within layout pattern strings. Note that each conversion qualifier starts with a percent sign (%).

Pattern Layout Conversion Characters	
Conversion Character	***Description***
%c	Category (name of logger) issuing the event.
%D	Date/time of the logging event. The date can be formatted using `%D[format]` where valid symbols for `[format]` are as follows: %Y: year, %m: month, %d: day, %H: hours, %M: minutes, %S: seconds, %%q: milliseconds.
%l	Location of the logging request (method, file name and line number). WARNING: This qualifier will severely impede performance!
%L	Line number of the logging request. WARNING: This qualifier will severely impede performance!
%m	Message of the log.
%n	New line.
%p	Priority of the logging event (logging level).
%r	Timestamp in milliseconds from the start of program until the creation of the logging event.
%t	Thread from which the logging request was made.

Step 3 (optional): Set child-specific behaviour

Logging configuration is performed at the root level, but any action can be overwritten by any “child” logger lower in the hierarchy. This way, you can assign a different logging level and logging format to a specific node in the logger hierarchy. Note that any log statement meeting the minimal logging level selected **will be forwarded to all appenders assigned to that logger** (directly or inherited from higher up in the logger hierarchy). For example, you might use this methodology to direct the log output to both the Windows Event Viewer *and* to a rolling file.

Dynamic call logging

The logger can be configured to redirect information about one particular call to a dedicated file. This “per call” information is actually **duplicated** from the current appenders and written to a unique file in the system. The call logger appender is in reality a dynamic appender, similar to a *log4cplus::FileAppender*, except that the output file name is determined dynamically based on the call ID.

The call logger is enabled by default in the *.properties* files:

```
# Call Logging
log4cplus.appender.CALL_LOG_APPENDER=log4cplus::NullAppender
log4cplus.appender.CALL_LOG_APPENDER.Directory=C:\Program Files\Sangoma
NetBorder Platform 2.0\logs\call-logs
log4cplus.appender.CALL_LOG_APPENDER.ImmediateFlush=true
log4cplus.appender.CALL_LOG_APPENDER.layout=log4cplus::PatternLayout
log4cplus.appender.CALL_LOG_APPENDER.layout.ConversionPattern=%D{%Y-%m-
%d %H:%M:%S:%%q} [%t] %p - %c : %m%n
# Set to true if you want a directory structure with
# year/month/day/hour for your call logs
netborder.infra.CallLogger.dateTimeDirectory=true
```

In the example above, *log4cplus.appender.CALL_LOG_APPENDER.Directory* is set to a valid output directory with write permission; therefore, call logs are written to that directory by default. Note also that the parameter netborder*.infra.CallLogger.dateTimeDirectory* is set to “true”. This indicates that a sub-directory structure with a year, month, day and hour hierarchy will be created under the directory specified by the *Directory* property.

The *CALL_LOG_APPENDER* is different from other appenders because it cannot be attached to a logger at initialization time via a properties file. In fact, there is one *CALL_LOG_APPENDER* for each call that takes place, thus the terminology “dynamic appender”. For this reason, only code that has been specifically instrumented for call logging can use this appender.

Syslog integration

Logging to *syslog* involves adding the *REMOTE_SYSLOG_APPENDER* to the root logger and enabling network logging in *syslogd*.

Step 1: Add a Syslog appender

To use the remote syslog appender, you would amend the *.properties* file(s) as follows:

```
log4cplus.rootLogger=WARN, REMOTE_SYSLOG_APPENDER
log4cplus.appender.REMOTE_SYSLOG_APPENDER=
                              netborder::RemoteSyslogAppender
log4cplus.appender.REMOTE_SYSLOG_APPENDER.layout=
                              log4cplus::PatternLayout
log4cplus.appender.REMOTE_SYSLOG_APPENDER.layout.ConversionPattern=
%D{%H:%M:%S:%%q} [%t] %p - %c : %m%n
log4cplus.appender.REMOTE_SYSLOG_APPENDER.Hostname=hostname
log4cplus.appender.REMOTE_SYSLOG_APPENDER.Port=514
log4cplus.appender.REMOTE_SYSLOG_APPENDER.Facility=8
```

The last two settings are optional and have default values of 514 and 8, respectively: 514 is the default syslog *Port*, and the *Facility* property represents the source of the message (8 stands for LOG_USER or random user-level messages).

Step 2: Enable network logging in syslogd

To enable network logging in *syslogd*, you must make certain it is launched with the `-r` option. This option will tell *syslogd* to accept logging messages from remote hosts.

Appendix D: SIP response codes

This appendix contains a list of SIP response codes you may encounter while using the NetBorder Call Analyzer.

For a complete list of SIP response codes, refer to Section 21 of RFC 3261.

Typically, a server sends a SIP response to a client to indicate the status of a SIP request that the client previously sent to the server. SIP responses are numbered from 100 to 600, and grouped in classes as follows:

- **1XX:** Indicates informational or provisional status, which should be followed by another response.
- **2XX:** Indicates successful processing of the SIP request.
- **3XX:** Indicates that the SIP request needs to be redirected. Further action needs to be taken in order to complete the request.
- **4XX:** Indicates client error. The request either contains bad syntax or cannot be fulfilled at this server.
- **5XX:** Indicates server error. The server failed to fulfill an apparently valid request.
- **6XX:** Indicates global failure. The request cannot be fulfilled at any server.

The following table contains the most frequent SIP response codes you will encounter while using the NetBorder Call Analyzer.

SIP Response Codes		
Response Code	***Response Name***	***Explanation***
1XX Provisional		
100	Trying	This response indicates that the request has been received and that some unspecified action is being taken on behalf of this call (for example, a database is being consulted). This response, like all other provisional responses, stops retransmissions of an INVITE by a UAC.
180	Ringing	The UA receiving the INVITE is trying to alert the user.
183	Session Progress	This response is used to convey information about the progress of the call that is not otherwise classified. The Reason-Phrase, header fields, or message body MAY be used to convey more details about the call progress.
2XX Successful		
200	OK	The request has succeeded.
202	Accepted	This response indicates that the request has been accepted for processing, but the processing has not been completed. The request might or might not eventually be acted upon, as it might be disallowed when processing actually takes place. Used primarily for referrals.
4XX Client-Error		
404	Not found	The phone number or extension included in the INVITE or REFER URI is either badly formed or unsupported/unreachable.
408	Request Timeout	A connection failed to be reached within the specified timeframe (no answer from remote party).

SIP Response Codes		
Response Code	***Response Name***	***Explanation***
480	Temporarily Unavailable	The callee's end system was contacted successfully but the callee is currently unavailable (for example, is not logged in, is logged in but in a state that precludes communication with the callee, or has activated the “do not disturb” feature). The response MAY indicate a better time to call in the Retry-After header field.
486	Busy Here	The callee's end system was contacted successfully, but the callee is busy (or currently not willing or able to take additional calls at this end system). The response MAY indicate a better time to call in the Retry-After header field.
488	Not Acceptable Here	Same as 606 (“Not Acceptable”), but applies only to the specific resource addressed by the Request-URI. The request may succeed elsewhere.
5XX Server-Error		
500	Internal Server Error	An internal error has occurred.
503	Service Unavailable	An error occurred at the physical level.
504	Server Timeout	An “alerting” message was not received within the allowable timeframe.
6XX Global Failure		
600	Busy Everywhere	The callee's end system was contacted successfully but the callee is busy and does not wish to take the call at this time. The response MAY indicate a better time to call in the Retry-After header field. This status response is returned only if the client knows that no other end point (such as a voice mail system) will answer the request.

SIP Response Codes		
Response Code	***Response Name***	***Explanation***
603	Decline	The user dropped the call. Typically, this happens when the callee hangs up before media detection is complete. It can also be caused by the maximum licensing capacity being reached. Look for *603 Decline (Licensing capacity has been reached)* in file: *[NETBORDER_INSTALLDIR]/* logs/call-analyzer-service.out
606	Not Acceptable	The user's agent was contacted successfully but some aspects of the session description, such as the requested media, bandwidth, or addressing style, were not acceptable. This status response means that the user wishes to communicate, but cannot adequately support the session described. The response MAY contain a list of reasons in a Warning header field explaining why the session described cannot be supported.

Appendix E: Sample SIP messages

This appendix contains sample call flows, and an example of the SIP messages received and generated by the NetBorder Call Analyzer in the process of obtaining a NCA result.

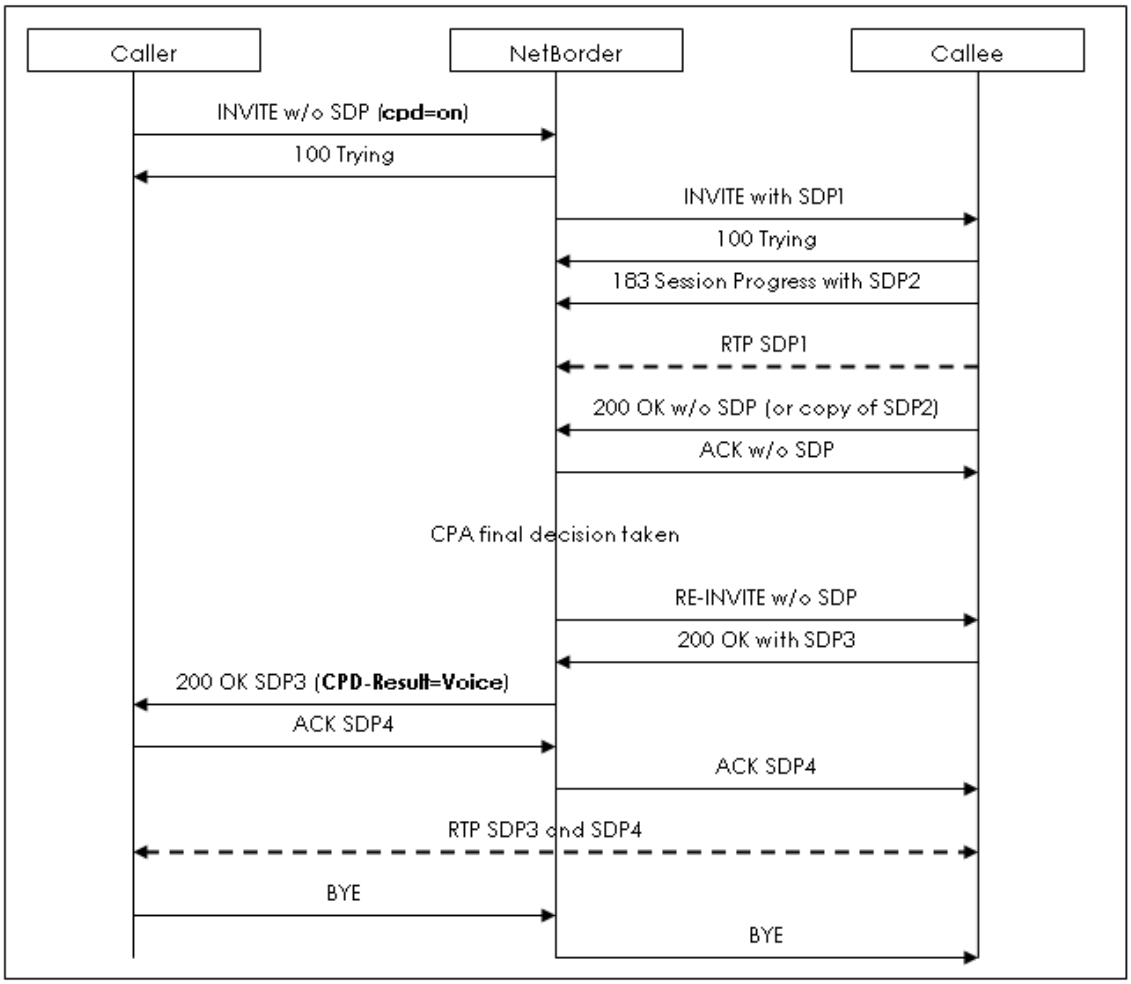
This appendix contains the following topics:

Sample call flows

A SIP transaction commences when a participant is invited to a call. The first INVITE is sent with or without an SDP (Session Description Protocol) body.

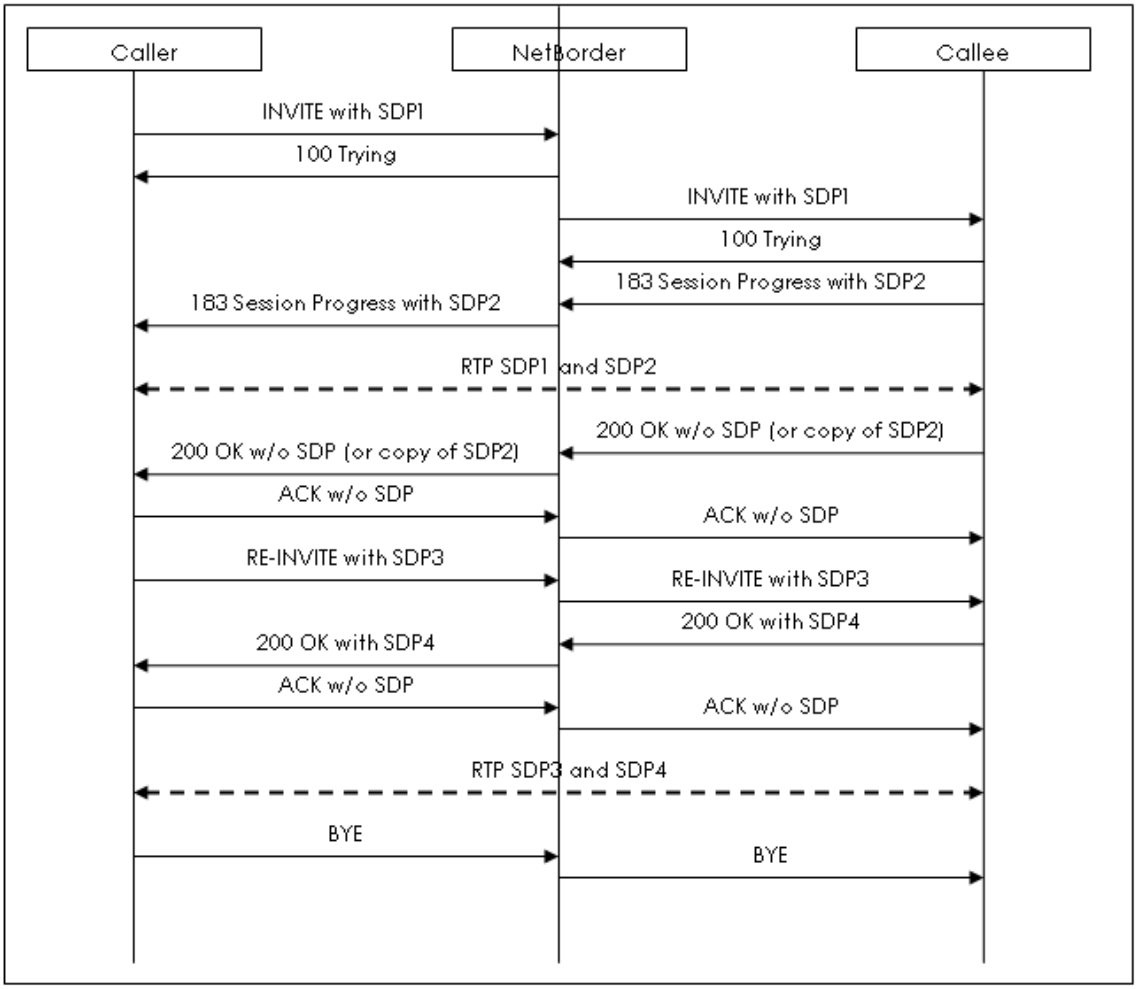
In the context of a Contact Center, if the first INVITE contains an SDP, then a RE-INVITE is sent by the dialer (or caller) with the SDP of an available agent. On the other hand, if the first INVITE does not contain an SDP body, the ACK message will contain the SDP of the agent.

The diagram below presents a scenario in which the first INVITE does **not** contain an SDP body. Note that call progress analysis has been enabled (**cpd=on**).



Modified versions of the first SDP are labelled SDP2, SDP3, and so on.

This second example depicts a sample call flow in which call progress analysis has **not** been enabled, and the first INVITE contains an SDP body.



Sample SIP messages

This section presents an example of the SIP messages received and generated by the NetBorder Call Analyzer in the process of obtaining a NCA result.

Note that the NetBorder Call Analyzer sits in the middle between the caller and the callee. Its role is the orchestration of the SIP messages between the UAs involved.

>>> INVITE received by the caller

```
INVITE sip:1024@192.168.11.103:5062;cpd=on SIP/2.0
Via: SIP/2.0/UDP
192.168.11.156:5060;branch=z9hG4bK233E73631D7A9A90B05D8C5A02983766;rport=5060
Max-Forwards: 70
Contact: <sip:Username@192.168.11.156:5060;transport=udp>
To: <sip:1024@192.168.11.103:5062>
From: "Name"<sip:Username@defaultproxy:5060>;tag=660A72622E4A7F9B9A839A2B8B9381FD
Call-ID: FC735409EBE21C394E69C4AD800FBF01@192.168.11.156:5062
CSeq: 1 INVITE
Session-Expires: 1800;refresher=uac
Content-Type: application/sdp
Supported: timer, replaces
User-Agent: Kapanga Softphone Desktop 1.00/2163e+1161886252_001372BDBCE2
Content-Length: 323

v=0
o=Username 1190134451 1190750689 IN IP4 192.168.11.156
s=Kapanga [1190134451]
c=IN IP4 192.168.11.156
t=0 0
m=audio 5562 RTP/AVP 8 0 101
a=rtpmap:8 pcma/8000
a=sendrecv
a=silenceSupp:off - - -
a=rtcp:5563
a=maxptime:20
a=ptime:20
a=rtpmap:0 pcmu/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15,36
```

<<< 100 Trying sent to the caller

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP
192.168.11.156:5060;branch=z9hG4bK233E73631D7A9A90B05D8C5A02983766;rport=5060
To: <sip:1024@192.168.11.103:5062>;tag=0f7fe217
From: "Your Long
Name"<sip:Username@defaultproxy:5060>;tag=660A72622E4A7F9B9A839A2B8B9381FD
Call-ID: FC735409EBE21C394E69C4AD800FBF01@192.168.11.156:5062
CSeq: 1 INVITE
Content-Length: 0
```

<<< INVITE sent to the callee with the SDP of NetBorder

```
INVITE sip:1024@192.168.11.103:5061 SIP/2.0
Via: SIP/2.0/UDP 192.168.11.156:5062;branch=z9hG4bK-d87543-170fcc611560b358-1---
d87543-;rport
Max-Forwards: 70
Contact: <sip:NetBorder@192.168.11.156:5062>
To: <sip:1024@192.168.11.103:5061>
From: "Paraxip NetBorder"<sip:NetBorder@127.0.0.1:5062>;tag=ee58eb5f
Call-ID: MzBhM2M2MmQ1NzVkZTFkZDhjY2FjYWNiNTQ4ZTJkMDE.
CSeq: 1 INVITE
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, INFO
Content-Type: application/sdp
Content-Length: 193

v=0
o=- 1190750690 1190750690 IN IP4 192.168.11.156
s=Paraxip Media Session
c=IN IP4 192.168.11.156
t=0 0
m=audio 9000 RTP/AVP 0 8
a=recvonly
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
```

>>> 100 Trying received from the callee

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.11.156:5062;branch=z9hG4bK-d87543-170fcc611560b358-1---
d87543-;rport=5062
Contact: sip:ParaxipGateway@192.168.11.103:5062;transport=udp
To: <sip:1024@192.168.11.103:5062>;tag=ds-6784-8b3c33d9
From: "Paraxip NetBorder"<sip:NetBorder@127.0.0.1:5062>;tag=ee58eb5f
Call-ID: MzBhM2M2MmQ1NzVkZTFkZDhjY2FjYWNiNTQ4ZTJkMDE.
CSeq: 1 INVITE
```

```
Server: Paraxip Gateway/2.2.0
Content-Length: 0

>>> 180 Ringing received from callee (with SDP, then in early media mode)

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.168.11.156:5062;branch=z9hG4bK-d87543-170fcc611560b358-1---
d87543-;rport=5062
Contact: sip:ParaxipGateway@192.168.11.103:5061;transport=udp
To: <sip:1024@192.168.11.103:5061>;tag=ds-6784-8b3c33d9
From: "Paraxip NetBorder"<sip:NetBorder@127.0.0.1:5062>;tag=ee58eb5f
Call-ID: MzBhM2M2MmQ1NzVkZTFkZDhjY2FjYWNiNTQ4ZTJkMDE.
CSeq: 1 INVITE
Content-Type: application/sdp
Server: Paraxip Gateway/2.2.0
Content-Length: 180

v=0
o=Paraxip-Tech 1190750737 1190750737 IN IP4 192.168.11.103
s=SIP Call
c=IN IP4 192.168.11.103
t=0 0
m=audio 49152 RTP/AVP 0
a=rtpmap:0 pcmu/8000
a=ptime:20
a=sendonly

<<< 180 Ringing sent back to the caller

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP
192.168.11.156:5060;branch=z9hG4bK233E73631D7A9A90B05D8C5A02983766;rport=5060
Contact: "Paraxip NetBorder"<sip:NetBorder@127.0.0.1:5062>
To: <sip:1024@192.168.11.103:5061>;tag=0f7fe217
From: "Name"<sip:Username@defaultproxy:5060>;tag=660A72622E4A7F9B9A839A2B8B9381FD
Call-ID: FC735409EBE21C394E69C4AD800FBF01@192.168.11.156:5062
CSeq: 1 INVITE
Content-Length: 0

>>> 200 OK received from the callee (callee answer)

SIP/2.0 200 Ok
Via: SIP/2.0/UDP 192.168.11.156:5062;branch=z9hG4bK-d87543-170fcc611560b358-1---
d87543-;rport=5062
Contact: sip:ParaxipGateway@192.168.11.103:5061;transport=udp
To: <sip:1024@192.168.11.103:5061>;tag=ds-6784-8b3c33d9
```

```
From: "Paraxip NetBorder"<sip:NetBorder@127.0.0.1:5062>;tag=ee58eb5f
Call-ID: MzBhM2M2MmQ1NzVkZTFkZDhjY2FjYWNiNTQ4ZTJkMDE.
CSeq: 1 INVITE
Content-Type: application/sdp
Server: Paraxip Gateway/2.2.0
Content-Length: 180

v=0
o=Paraxip-Tech 1190750737 1190750738 IN IP4 192.168.11.103
s=SIP Call
c=IN IP4 192.168.11.103
t=0 0
m=audio 49152 RTP/AVP 0
a=rtpmap:0 pcmu/8000
a=ptime:20
a=sendonly
```

<<< ACK sent to the callee for the 200 OK

```
ACK sip:ParaxipGateway@192.168.11.103:5061;transport=udp SIP/2.0
Via: SIP/2.0/UDP 192.168.11.156:5062;branch=z9hG4bK-d87543-f32cda4907194b6b-1---
d87543-;rport
Max-Forwards: 70
Contact: <sip:NetBorder@192.168.11.156:5062>
To: <sip:1024@192.168.11.103:5061>;tag=ds-6784-8b3c33d9
From: "Paraxip NetBorder"<sip:NetBorder@127.0.0.1:5062>;tag=ee58eb5f
Call-ID: MzBhM2M2MmQ1NzVkZTFkZDhjY2FjYWNiNTQ4ZTJkMDE.
CSeq: 1 ACK
Content-Length: 0
```

<<< RE-INVITE sent to the callee with the caller SDP

```
INVITE sip:ParaxipGateway@192.168.11.103:5061;transport=udp SIP/2.0
Via: SIP/2.0/UDP 192.168.11.156:5062;branch=z9hG4bK-d87543-6964a8270b272a38-1---
d87543-;rport
Max-Forwards: 70
Contact: <sip:NetBorder@192.168.11.156:5062>
To: <sip:1024@192.168.11.103:5061>;tag=ds-6784-8b3c33d9
From: "Paraxip NetBorder"<sip:NetBorder@127.0.0.1:5062>;tag=ee58eb5f
Call-ID: MzBhM2M2MmQ1NzVkZTFkZDhjY2FjYWNiNTQ4ZTJkMDE.
CSeq: 2 INVITE
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, INFO
Content-Type: application/sdp
Content-Length: 323

v=0
```

```
o=Username 1190134451 1190750689 IN IP4 192.168.11.156
s=Kapanga [1190134451]
c=IN IP4 192.168.11.156
t=0 0
m=audio 5562 RTP/AVP 8 0 101
a=fmtp:101 0-15,36
a=maxptime:20
a=ptime:20
a=rtcp:5563
a=rtpmap:8 pcma/8000
a=rtpmap:0 pcmu/8000
a=rtpmap:101 telephone-event/8000
a=sendrecv
a=silenceSupp:off - - -
```

>>> 200 OK received by the callee for the RE-INVITE

```
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 192.168.11.156:5062;branch=z9hG4bK-d87543-6964a8270b272a38-1---
d87543-;rport=5062
Contact: sip:ParaxipGateway@192.168.11.103:5061;transport=udp
To: <sip:1024@192.168.11.103:5061>;tag=ds-6784-8b3c33d9
From: "Paraxip NetBorder"<sip:NetBorder@127.0.0.1:5062>;tag=ee58eb5f
Call-ID: MzBhM2M2MmQ1NzVkZTFkZDhjY2FjYWNiNTQ4ZTJkMDE.
CSeq: 2 INVITE
Content-Type: application/sdp
Content-Length: 236

v=0
o=Paraxip-Tech 1190750737 1190750739 IN IP4 192.168.11.103
s=SIP Call
c=IN IP4 192.168.11.103
t=0 0
m=audio 49152 RTP/AVP 0 101
a=rtpmap:0 pcmu/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=sendrecv
```

<<< ACK sent to the callee

```
ACK sip:ParaxipGateway@192.168.11.103:5061;transport=udp SIP/2.0
Via: SIP/2.0/UDP 192.168.11.156:5062;branch=z9hG4bK-d87543-60077a36f9536001-1---
d87543-;rport
Max-Forwards: 70
```

Contact: <sip:NetBorder@192.168.11.156:5062>
To: <sip:1024@192.168.11.103:5061>;tag=ds-6784-8b3c33d9
From: "Paraxip NetBorder"<sip:NetBorder@127.0.0.1:5062>;tag=ee58eb5f
Call-ID: MzBhM2M2MmQ1NzVkZTFkZDhjY2FjYWNiNTQ4ZTJkMDE.
CSeq: 2 ACK
Content-Length: 0

<<< 200 OK sent to the caller (answer the call) with the callee SDP

SIP/2.0 200 OK
Via: SIP/2.0/UDP
192.168.11.156:5060;branch=z9hG4bK233E73631D7A9A90B05D8C5A02983766;rport=5060
Contact: "Paraxip NetBorder"<sip:NetBorder@127.0.0.1:5062>
To: <sip:1024@192.168.11.103:5061>;tag=0f7fe217
From: "Your Long
Name"<sip:Username@defaultproxy:5060>;tag=660A72622E4A7F9B9A839A2B8B9381FD
Call-ID: FC735409EBE21C394E69C4AD800FBF01@192.168.11.156:5062
CSeq: 1 INVITE
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, INFO
Content-Type: application/sdp
Content-Length: 236
CPD-Result: Voice

v=0
o=Paraxip-Tech 1190750737 1190750739 IN IP4 192.168.11.103
s=SIP Call
c=IN IP4 192.168.11.103
t=0 0
m=audio 49152 RTP/AVP 0 101
a=fmtp:101 0-16
a=ptime:20
a=rtpmap:0 pcmu/8000
a=rtpmap:101 telephone-event/8000
a=sendrecv

>>> ACK received from caller for the 200 OK

ACK sip:NetBorder@127.0.0.1:5062 SIP/2.0
Via: SIP/2.0/UDP
192.168.11.156:5060;branch=z9hG4bK22EF4C58F79F7D3085BE267479EE0877;rport=5060
Max-Forwards: 70
Contact: <sip:Username@192.168.11.156:5060;transport=udp>
To: <sip:1024@192.168.11.103:5061>;tag=0f7fe217
From: "Your Long
Name"<sip:Username@defaultproxy:5060>;tag=660A72622E4A7F9B9A839A2B8B9381FD
Call-ID: FC735409EBE21C394E69C4AD800FBF01@192.168.11.156:5062

```
CSeq: 1 ACK
User-Agent: Kapanga Softphone Desktop 1.00/2163e+1161886252_001372BDBCE2
Content-Length: 0
```

>>> BYE received from the callee

```
BYE sip:NetBorder@192.168.11.156:5062;transport=udp SIP/2.0
Via: SIP/2.0/UDP 192.168.11.103:5061;branch=z9hG4bKb40d23cb-6ba2-11dc-a6a1-9f96e950026a
Max-Forwards: 70
Contact: <sip:192.168.11.103:5061;transport=udp>
To: "Paraxip NetBorder"<sip:NetBorder@127.0.0.1:5062>;tag=ee58eb5f
From: <sip:1024@192.168.11.103:5061>;tag=ds-6784-8b3c33d9
Call-ID: MzBhM2M2MmQ1NzVkZTFkZDhjY2FjYWNiNTQ4ZTJkMDE.
CSeq: 2 BYE
Content-Length: 0
```

<<< 200 OK sent to the callee for the BYE

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.11.103:5061;branch=z9hG4bKb40d23cb-6ba2-11dc-a6a1-9f96e950026a
Contact: <sip:NetBorder@192.168.11.156:5062>
To: "Paraxip NetBorder"<sip:NetBorder@127.0.0.1:5062>;tag=ee58eb5f
From: <sip:1024@192.168.11.103:5061>;tag=ds-6784-8b3c33d9
Call-ID: MzBhM2M2MmQ1NzVkZTFkZDhjY2FjYWNiNTQ4ZTJkMDE.
CSeq: 2 BYE
Content-Length: 0
```

>>> BYE sent to the caller

```
BYE sip:Username@192.168.11.156:5060;transport=udp SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:5062;branch=z9hG4bK-d87543-9457d20c5602cd16-1---d87543-;rport
Max-Forwards: 70
Contact: "Paraxip NetBorder"<sip:NetBorder@127.0.0.1:5062>
To: "Your Long
Name"<sip:Username@defaultproxy:5060>;tag=660A72622E4A7F9B9A839A2B8B9381FD
From: <sip:1024@192.168.11.103:5061>;tag=0f7fe217
Call-ID: FC735409EBE21C394E69C4AD800FBF01@192.168.11.156:5062
CSeq: 2 BYE
Content-Length: 0
```

<<< 200 OK received from the caller for the BYE

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 127.0.0.1:5062;branch=z9hG4bK-d87543-9457d20c5602cd16-1---
d87543-;received=192.168.11.156
Contact: <sip:Username@192.168.11.156:5060;transport=udp>
```

```
To: "Your Long
Name"<sip:Username@defaultproxy:5060>;tag=660A72622E4A7F9B9A839A2B8B9381FD
From: <sip:1024@192.168.11.103:5061>;tag=0f7fe217
Call-ID: FC735409EBE21C394E69C4AD800FBF01@192.168.11.156:5062
CSeq: 2 BYE
Supported: timer, replaces
User-Agent: Kapanga Softphone Desktop 1.00/2163e+1161886252_001372BDBCE2
Content-Length: 0
```