Sangoma Technologies Corporation

Netborder Express Gateway

User Guide

v1.6.0.
5 September 2008

Version: 1.6.0

Table of Contents

About this document

The Netborder Express Gateway is an open, software-based *VoIP (Voice over Internet Protocol)* gateway product that provides a comprehensive and highly flexible bridge between the traditional telephony network and IP-based platforms and applications.

Audience

This document is intended for application developers and system administrators who manage and interface with the Netborder Express Gateway.

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 telephony and data requirements of your VoIP system.
- You have a working knowledge of Windows operating systems, the Internet, and graphical user interfaces.
- You are prepared to use Sangoma telephony boards to connect the Gateway to the *Public Switched Telephone Network (PSTN)*.

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 Express Gateway as well as an explanation of the system architecture.
Chapter 2	Installation	Describes how to install (and uninstall) both the Gateway software and the Sangoma telephony boards.
Chapter 3	Getting started	Gets you started using the Gateway by making a SIP to PSTN call and a PSTN to SIP call.
Chapter 4	Operating the Gateway	Describes how to operate the Gateway and how to monitor its performance using the Web Interface.
Chapter 5	PSTN	Describes how to configure the parameters used to connect the Gateway to a traditional telephony TDM network.
Chapter 6	SIP and VoIP	Describes how to configure SIP and RTP, and explains how the Gateway maps its operations to conform to the SIP standard.
Chapter 7	Routing rules	Explains the function of routing rules and describes how you can create, delete, or modify rules.
Chapter 8	Engineering guidelines	Provides IP network and telephony considerations.
Chapter 9	Troubleshooting	Provides solutions to key troubleshooting issues.
Appendix A	Glossary	Contains a list of abbreviations and acronyms used in this guide.

Section	Title	Description
Appendix B	Configuration parameters	Provides a comprehensive list of parameters, with a brief description, for the main configuration files used by the Gateway.
Appendix C	Logging configuration	Contains general information about logging and logging configuration.
Appendix D	Sangoma boards	Provides a description of the Sangoma boards currently supported by the Gateway.
Appendix E	PSTN configuration file examples	Contains some examples of PSTN configuration file.
Appendix F	Modify the Microsoft Windows driver signing options	Describes how to modify windows options to prevent Windows to display a dialogue box about unsigned drivers.

Related documentation

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

- *Netborder Express Gateway 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 Corporation website at http://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.

Regular expressions

The Gateway's powerful Routing Engine makes abundant use of regular expressions, which are a popular means of representing and parsing text strings. Many online tutorials are available on the subject of regular expressions, including the following at http://perldoc.perl.org/:

- http://perldoc.perl.org/perlretut.html.

SIP RFCs

In March 1999, SIP 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 Express Gateway 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 “Netborder Express Gateway” Service, 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: *[GATEWAY_HOME]\config\gw.properties*

  where *[GATEWAY_HOME]* is the root folder of the installation (for example, *C:\Program Files\Netborder\Express\Gateway\config\gw.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 Express Gateway offers a flexible and powerful software solution for building a complete *Voice-over-IP (VoIP)* gateway. The software package is composed of the executable files, configuration files, product documentation, and samples required to assemble and operate a complete, yet customizable, VoIP gateway based on commercial hardware.

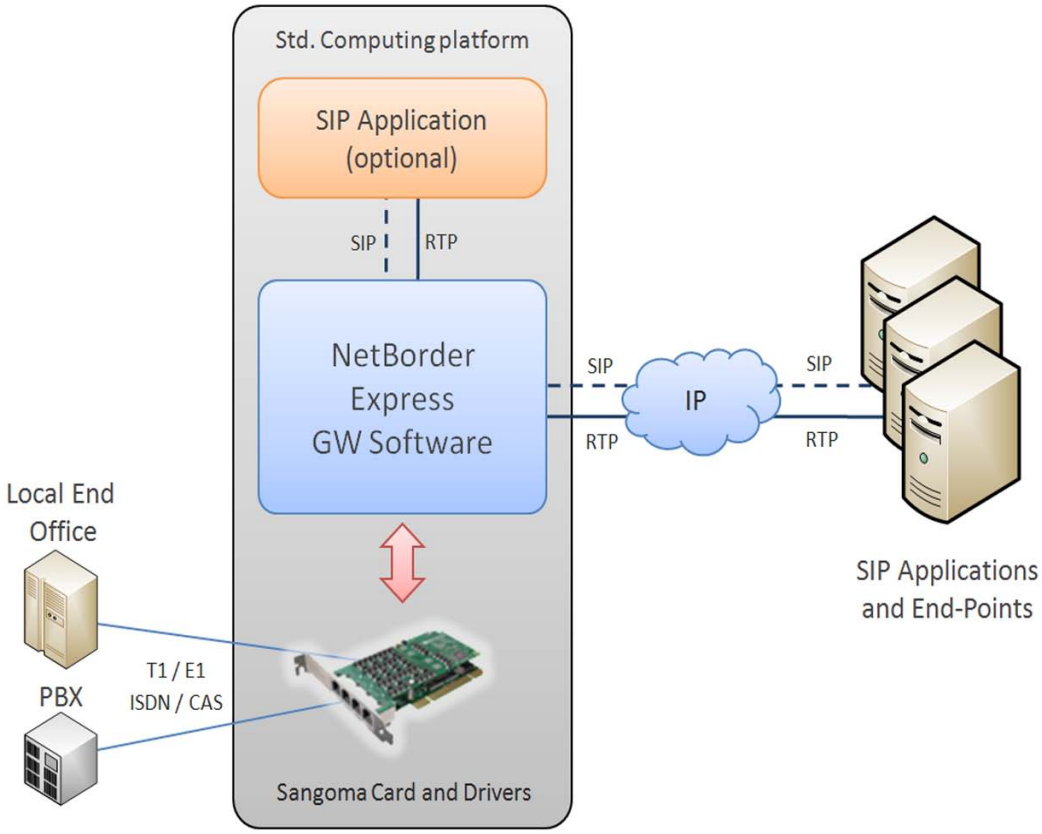
The Netborder Express Gateway provides a bridge between traditional telephony equipment and IP telephony applications such as SIP phones or SIP-based media servers. The Gateway enables transparent operations in an hybrid environment, mixing IP and traditional telephony equipment and applications.

This chapter contains:

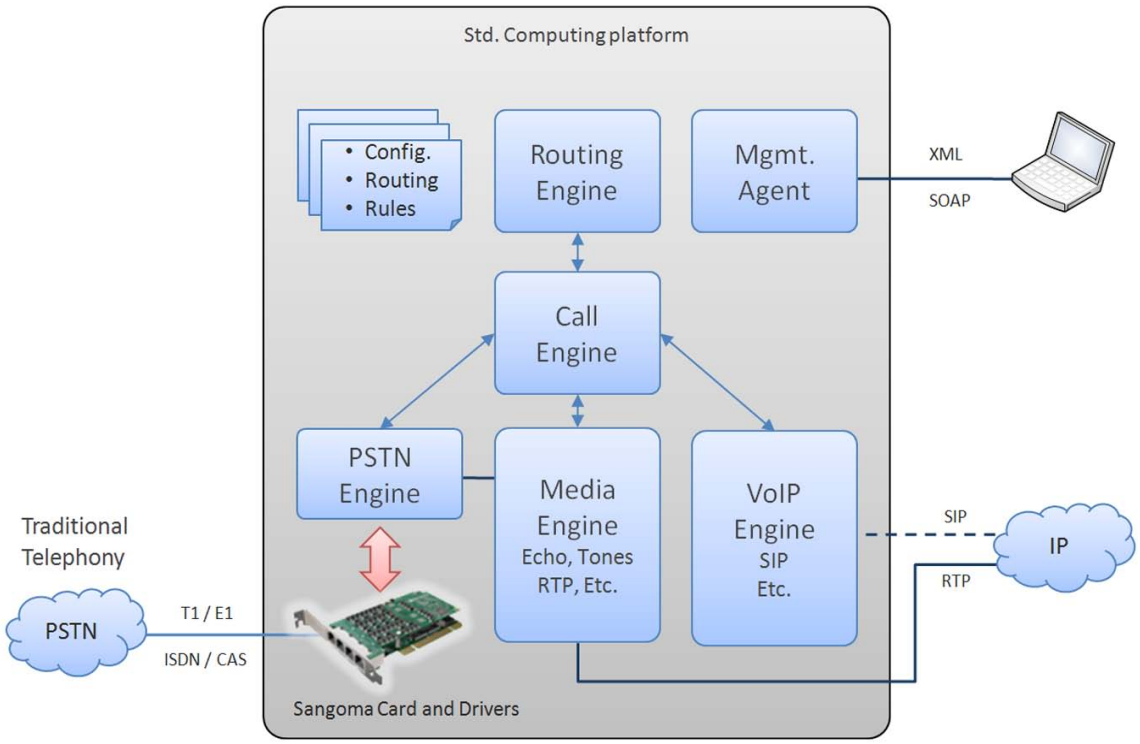
System architecture

The Gateway is installed on a server equipped with supported telephony and media processing cards. The telephony cards are connected to the enterprise *PBX (Private Branch Exchange)* or directly to Central Office lines. The Gateway ensures SIP connectivity to VoIP applications running on the same host and/or on the same local area network.

The figure below depicts the overall system architecture.



The product's high-level software architecture is illustrated and described below.



PSTN Engine and VoIP Engine

The PSTN Engine and VoIP Engine together control the traditional telephony and the IP telephony protocols. This includes controlling the state machines of the PSTN and VoIP call legs, and controlling the interface to the telephony board and SIP signaling stack.

Media Engine

The Media Engine is the entity controlling all of the media processing functions, including the "brokerage" of audio information between the PSTN Engine and the VoIP Engine. It also interfaces with the media processing hardware (if any) for functions such as echo cancellation and *RTP (Real-time Transport Protocol)* packetization.

Call Engine

The Call Engine is the heart of the Gateway application. It is where the concept of a call takes form. The Call Engine is capable of managing multi-legged calls, with the different legs being of arbitrary types (PSTN or VoIP).

Routing Engine

The Routing Engine is the component responsible for managing the routing rules, which indicate how to route inbound calls and transfer requests to the appropriate resources based on a number of parameters, such as *ANI (Automatic Number Identification)*, *DNIS (Dialed Number Information Service)*, Caller ID, or SIP *URI (Universal Resource Identifier)*.

The Routing engine is also used to massage the information contained in the input call parameters into the properties of the outgoing call. For example, SIP headers/parameters if the incoming call is SIP, or *ISDN (Integrated Services Digital Network)* parameters if the incoming call is PSTN.

Management Agent

The Management Agent is the component that provides an external operations interface to the different Netborder Express Gateway components.

Chapter 2: Installation

This section describes how to install (and uninstall) both the Netborder Express Gateway software and the Sangoma telephony boards.

For information on how to start using the Gateway once it 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.

Computer requirements

For this release of the Netborder Express Gateway, you will need the following:

- Microsoft® Windows XP or Windows 2003
- minimum Intel Core 2 Duo 2GHz with 2MB of L2 cache (4MB of L2 cache recommended) or equivalent
- minimum of 512MB of RAM (1024MB recommended) – see Provisioning below
- 100MB of available disk space (includes free space for logging)
- network connection to the VoIP gateway
- Microsoft® Internet Explorer (version 6.0 and higher) or Mozilla Firefox™ (version 2.0.0 and higher).

Provisioning

This section provides information on the Gateway service's typical CPU and memory usage for:

- 23, 30, 46, 60, 92, 120, 184, 240, 368 and 480 ports ISDN *PRI (Primary Rate Interface)* configurations using host-based RTP.

The benchmarks included here were established using the following environment:

- **Operating system:** Microsoft® Windows Server 2003 standard edition
- **CPU:** Intel Xeon X3210 (Quad Core), 2.13 GHz, 2x4MB of L2 cache
- **RAM:** 4GB, DDR2 667MHz
- **Sangoma Board:** 2 x A108d-X (PCI-Express, 8 spans per board and Hardware echo cancellation)

The test application placed incoming PSTN calls on half the available ports on the Gateway (targeting a VoiceXML application running on the same host), and received incoming PSTN calls (initiated by a SIP application through the Gateway) on the other half of the available ports, under continuous load. All RTP streams are configured to send 20ms G711 voice packets (RTCP is enabled).

The calls had an average duration of 3 minutes. The benchmarks were performed under production configurations, with minimal logging.

The **average** CPU time (as a percentage of available CPU time in the overall system) and physical RAM utilization (including Microsoft® Windows and Sangoma device drivers memory usage ~268MB) of the Gateway process on the test system, are provided below.

Number of Ports	*CPU (%)*	*RAM (MB)*
1 span		
23	2	284
30	3	290
2 spans		
46	4	300
60	6	312
4 spans		
92	8	332
120	10	360
8 spans		
184	12	396
240	16	448
16 spans		
368	22	496
480	28	588

WARNING: Remove any Sangoma device drivers software and hardware before proceeding with the installation. The Gateway has been validated with Sangoma device drivers **v6.0.5.8**. Other versions are **not** compatible with the Gateway.

Installing the software

The Netborder Express Gateway software installer is available from the Sangoma Technologies Corporation website. If you have not been provided with the necessary URL, contact Sangoma Technologies Corporation support at the following e-mail address:

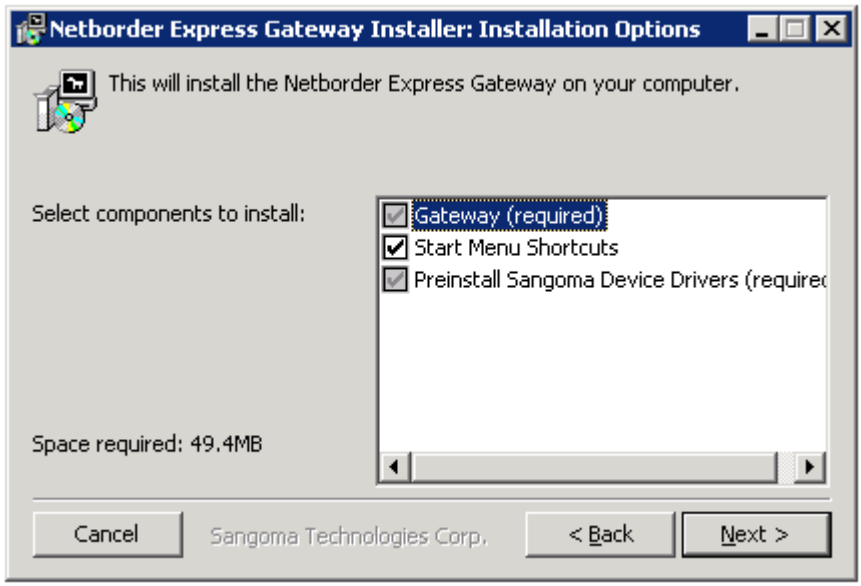
- support@sangoma.com

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

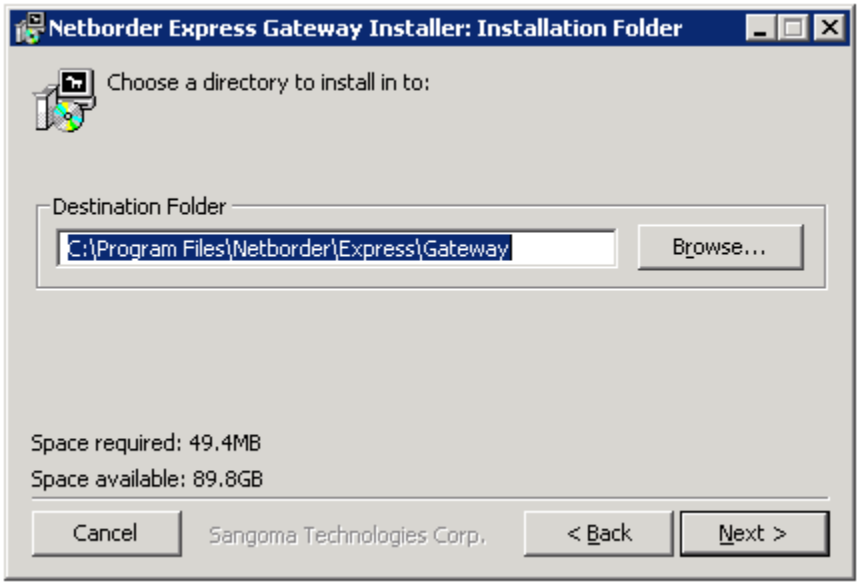
To install the Netborder Express Gateway software:

1. To begin the installation, double-click the following executable file:
   a) *NetborderExpressGatewaySetup1.6.0.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 Express Gateway software will not be installed.



3. Select the components you wish to install. The first and third options are required. To install Start Menu shortcuts (recommended), leave this option pre-selected.Start Menu shortcuts provide you with quick access to the Gateway Web Interface, logging information, configuration files, and with one-click access to essential functions such as Starting and Stopping the Gateway.
4. Click **Next**.
5. Set the home directory for the installation.



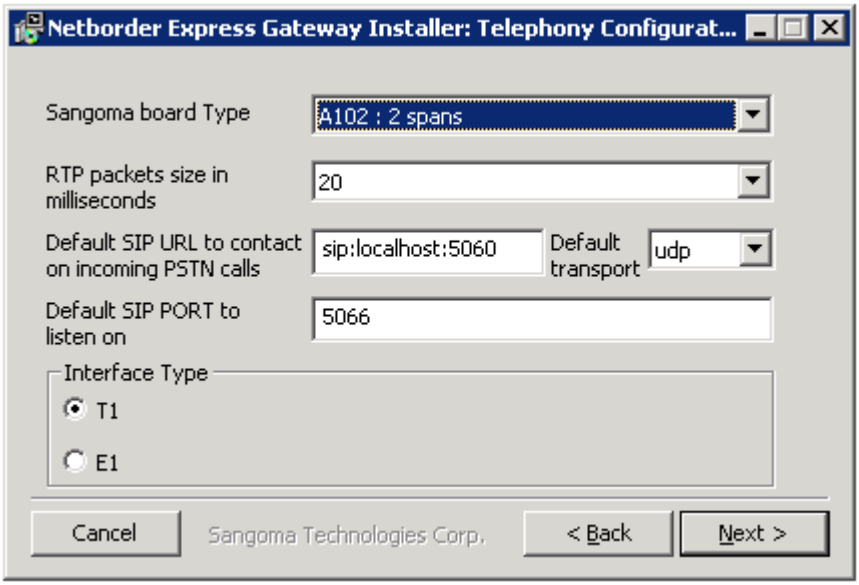
Do **one** of the following:

a) Use the default folder (*C:\Program Files\Netborder\Express\Gateway*).

b) Enter a new path.

c) Click **Browse...** to select a different directory.

6. Click **Next**.
7. Select the default telephony and routing configuration. Do the following:



a) Select the Sangoma board type. Your options are:

- A101 : 1 span
- A102 : 2 spans (default)
- A104 : 4 spans
- A108 : 8 spans
- 2 x A108 : 16 spans

b) Select the RTP packets size in milliseconds. Your options are:

- 10
- 20 (default)
- 30

c) Leave unchanged the Default SIP URL to contact on incoming PSTN calls.

By using this URL for SIP calls, you will be able to receive incoming PSTN calls once the installation is complete, without further configuration. (This URL can be changed later through the configuration files.)

a) Default transport to use when the gateway contact the default SIP URL. Your options are:

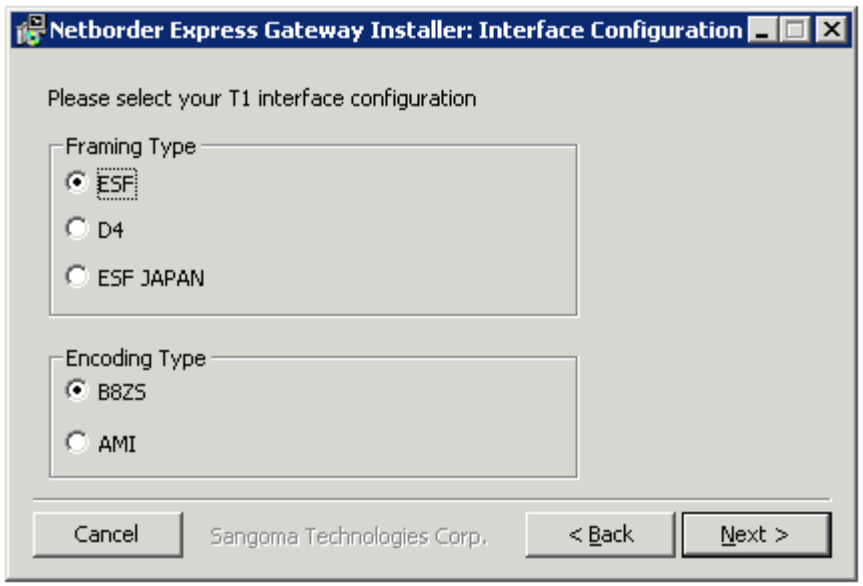
- udp: SIP messages will be transported in UDP packets.
- tcp: SIP messages will be transported in a TCP stream.

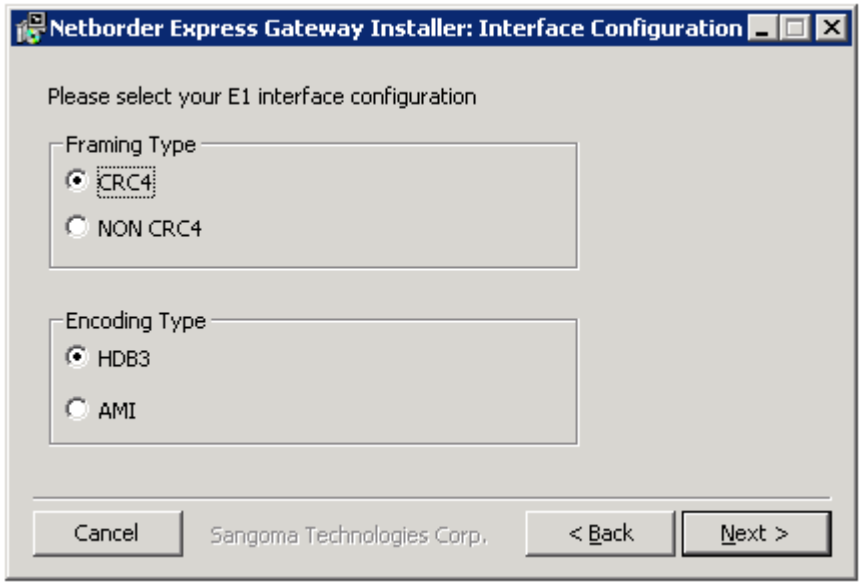
b) Default SIP port of the gateway.

d) Select the Interface type. Your options are:

- T1: Used in North America and Japan.
- E1: Used in Europe, Australia and most of the rest of the world.

8. Select the Framing Type and Encoding Type for your interface configuration. The options presented are dependent on the interface type you selected in the previous step (T1 or E1).





9. Configure your ISDN parameters.

   a) Select the ISDN switch variant. The options presented are dependent on the interface type you previously selected (T1 or E1). For T1 your options are:

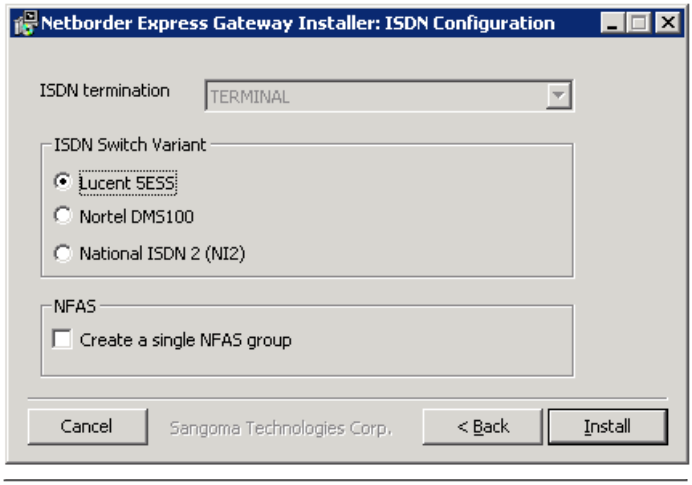
   - Lucent 5ESS (default)
   - Nortel DMS100

- National ISDN 2

For E1 your option is:

- Euro ISDN (default)

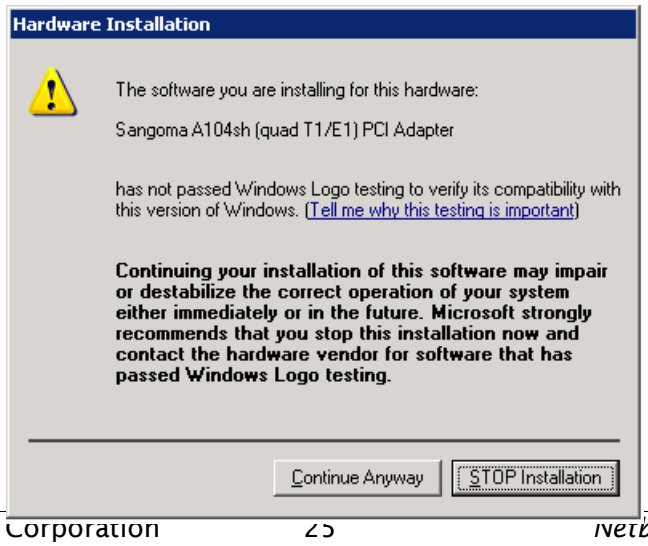
b) Select the Create a single NFAS group check box if you want to create a single NFAS group. This option presented when for T1 interface type only.



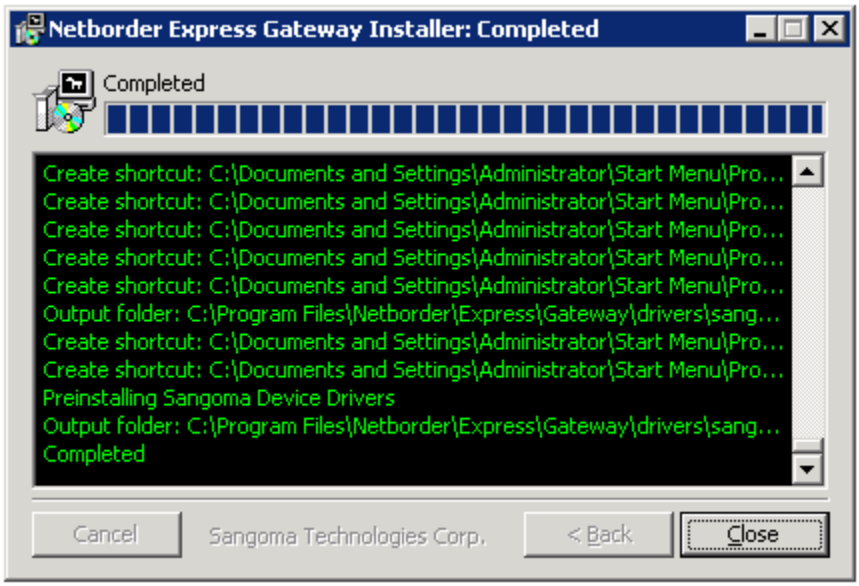
10. Click **Install** to complete the installation process.

    The required directories, subdirectories and files are installed on your system. For a complete list, see Directories and files installed by the Gateway software on page 27.

11. If a warning appears informing you that the software you are installing has not passed Windows logo testing, ignore the warning. Click **Continue Anyway**. Please refer to Appendix F: Modify the Microsoft Windows driver signing options to tell Windows not to ask for this confirmation.



12. If the “Digital Signature Not Found” warning appears, ignore the warning and continue the installation.
13. Once the installation is complete, install the Sangoma device drivers as described in section Installing the device driver software on page .



Directories and files installed by the Gateway software

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

Note that in the table below, [GATEWAY_HOME] is used to indicate the root folder of the installation (for example, *C:\Program Files\Netborder\Express\Gateway*).

List of Files Installed by the Gateway	
Folders and Files	***Description***
[GATEWAY_HOME]	
uninstall.exe	The root directory of the installation folder contains one file, the uninstall executable.
[GATEWAY_HOME]\bin	
dbghelp.dll DsAuthentication.dll DsMimeObject.dll DsSdpObject.dll DsSipLlApi.dll DsSipMlApi.dll DsSipObject.dll DsUtil.dll gw_webserver.conf Microsoft.VC80.DebugCRT.manifest mime.types msvcr80d.dll NTEventLogAppender.dll Netborder-ace.dll Netborder-boost-program-options.dll Netborder-boost-regex.dll Netborder-calleng.dll Netborder-curl.dll Netborder-geturl.dll Netborder-gsoap.dll Netborder-gw.dll Netborder-hostmedia.dll Netborder-infra.dll Netborder-jrtplib.dll Netborder-JRtpMediaEngine.dll Netborder-libdnet.dll Netborder-licensing.dll Netborder-log4cplus.dll	The *\bin* folder contains the executable files (EXE) required to run the Gateway, as well as the DLL (Dynamic Link Library) files, which contain code that is called upon when needed by the executable files. This folder also contains an icon image file, mime types file, configuration file for the web server embedded in the application, and a Microsoft application manifest file.

List of Files Installed by the Gateway	
Folders and Files	***Description***
Netborder-media.dll Netborder-net.dll Netborder-oam-cmd.exe Netborder-oam.dll Netborder-openssl-libeay32.dll Netborder-pstn-media.dll Netborder-pstn.dll Netborder-regexx.dll Netborder-rtp.dll Netborder-sangoma.dll Netborder-SangomaMediaEngine.dll Netborder-shttpd.dll Netborder-sip-client-reg.dll Netborder-sip-common.dll Netborder-sip-netif.dll Netborder-sip.dll Netborder-stlport45.dll Netborder-telesoft-netif.dll Netborder-telesoft-stack.dll Netborder-test.dll Netborder-thread.dll Netborder-vocalos-licensing.dll Netborder-vsm.exe Netborder-webserver.dll Netborder-win32-bugslayer-util.dll Netborder-xml.dll Netborder-zlib.dll shttpd.conf	
[GATEWAY_HOME]\bin\web	
help.html index.htm robots.txt routing-rules-edit.esp vital_signs.esp	The *bin**web* folder contains the files required for the Gateway Web Interface, including the required image files in the *[GATEWAY_HOME]**bin**web**files*\ subdirectory.
[GATEWAY_HOME]\config	
custom_tone_relay.conf dev-logger.properties gw.properties gw.properties install.properties netborder-express-gateway-license.txt netborder-express-gateway-license.txt.sig prod-logger.properties pstn-config.rng pstn-config.xml routing-rules.xml sip-client-registration.xml	The *config* folder contains the configuration files required by the Gateway. This folder also contains the license TXT file, which lists information about your existing license, including number of channels (ports) permitted, version number of the software, and expiry date of the license.

List of Files Installed by the Gateway	
Folders and Files	***Description***
[GATEWAY_HOME]\doc	
open-source-license.zip rules.dtd Service.wsdl TDM_Gateway_User_Guide.pdf Release_Notes.pdf	The *doc* folder contains documentation relevant to the Gateway, including the following: ● the Gateway's open source software license* ● Release Notes* ● the Document Type Definition for the *routing-rules.xml* file. ● web services interface description in Web Services Description Language file (WSDL) format ● this user guide*. * These documents are easily accessible from the Gateway Web Interface via the Help tab.
[GATEWAY_HOME]\logs\call-logs	The *logs**call-logs* folder contains all the log files. This folder will be empty immediately after installation.
[GATEWAY_HOME]\drivers\sangoma	
ec_files\ *hw_abstraction_driver*\ *utils*\	The *drivers**sangoma* folder and its subdirectories contain device drivers software, such as the installation files required for Sangoma, including for the first device driver (*SngBus.inf*) and for the individual interfaces required (*hw_abstraction_driver**sdladrv.inf*).
[GATEWAY_HOME]\tones	The *tones* folder contains sound files that are used to generate DTMF tones.

Installing/uninstalling the Sangoma board

This section describes how to install (and uninstall) a Sangoma board (T1 and E1) in the host system, as well as the required device drivers.

Before installing the board and drivers, please take note of the following:

- Make sure the Netborder Express Gateway software has been installed on the host system. See Installing the software on page 20.
- Stop the Gateway, if necessary. See Managing the service on page 76.
- Remove any pre-existing Sangoma AFT device drivers and/or hardware from the system. **The Gateway is compatible with Sangoma device drivers v6.0.5.8; other versions are incompatible.**

NOTE: The procedures and screen shots that follow are based on a Windows 2003 installation. If you are running Windows XP, the steps and accompanying screen shots may vary slightly.

NOTE: The procedures that follow require administrative privileges. Make sure to log on has an administrator.

NOTE: For administrators who install the device drivers from a remote desktop/terminal session, make sure to connect to the console session. Please refer to the /console option of the Microsoft Terminal Server Client application (mstsc.exe) under Windows XP Professional or Windows 2003.

Inserting the board in the chassis

This section provides procedures for installing a Sangoma board in the chassis of the host system.

CAUTION: Be sure to turn off the power supply of the host system before installing the board. Static electricity can damage sensitive electronic components. New boards are sealed in anti-static packaging before shipping. Before removing a board from its packaging, please ensure that you are grounded (for example, touch a well-grounded object such as your PC chassis).

To install the Sangoma board in the chassis:

1. If you have not already done so, **make sure the power supply of the host system is turned off**.
2. Handling only the mounting bracket, insert the board into a PCI/PCI Express slot.
3. Make sure that the board is properly seated, and that the fasteners are tightened properly.
4. Plug in the A101/2/4/8 PCI/PCI Express adapter.
5. Turn on the system.

The “New Hardware Wizard” will start automatically. To install the required drivers, see Installing the device drivers on page 31.

Installing the device drivers

After installing the board and restarting the system, a wizard tool will appear indicating that new hardware has been detected on your system. The tool will ask for the board’s driver. Once the driver is installed, the New Hardware Wizard will appear once again, asking you to install the required interfaces, one at a time.

NOTE: On some systems the operating system may automatically install the device drivers without prompting the user. In such cases go to Step 3: Validating devices driver installation on page 38.

This section has been divided into tree steps:

- Step 1: Installing the driver for the board on page 32
- Step 2: Installing the drivers for each of the interfaces on page 36.
- Step 3: Validating devices driver installation on page 38.

Step 1: Installing the driver for the board

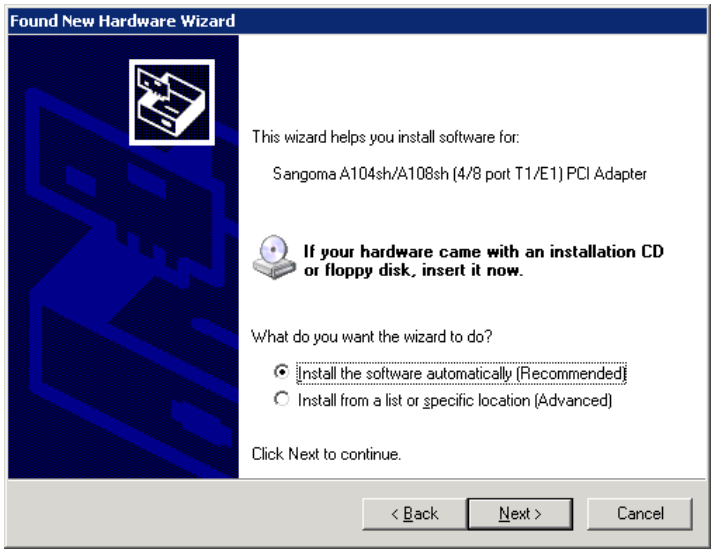
To install the device driver for the Sangoma board:

1. On the Welcome screen of the New Hardware Wizard, choose the "No, not this time" option to prevent Windows from searching for the required software through Windows Update.

NOTE: On some system, the Welcome to the Found New Hardware Wizard window may be shown partially and it is impossible to resize that windows. In such case, activate the Found New Hardware Wizard window by clicking anywhere in that window and press on the ESC key on your keyboard. This is will close the Found new Hardware window. Reboot the chassis and restart the procedure to step 1.

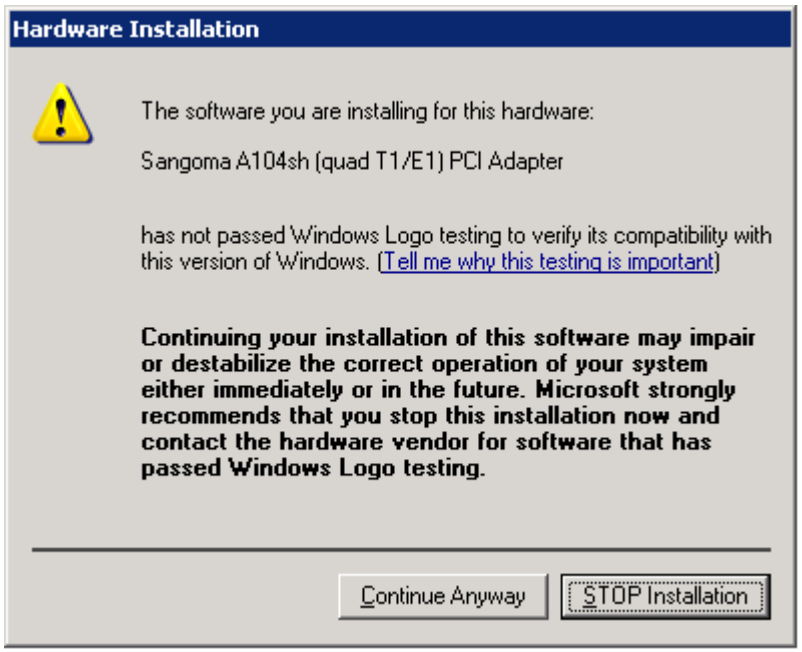
2. Click **Next**.

3. Select the option “Install software automatically (Recommended)”.



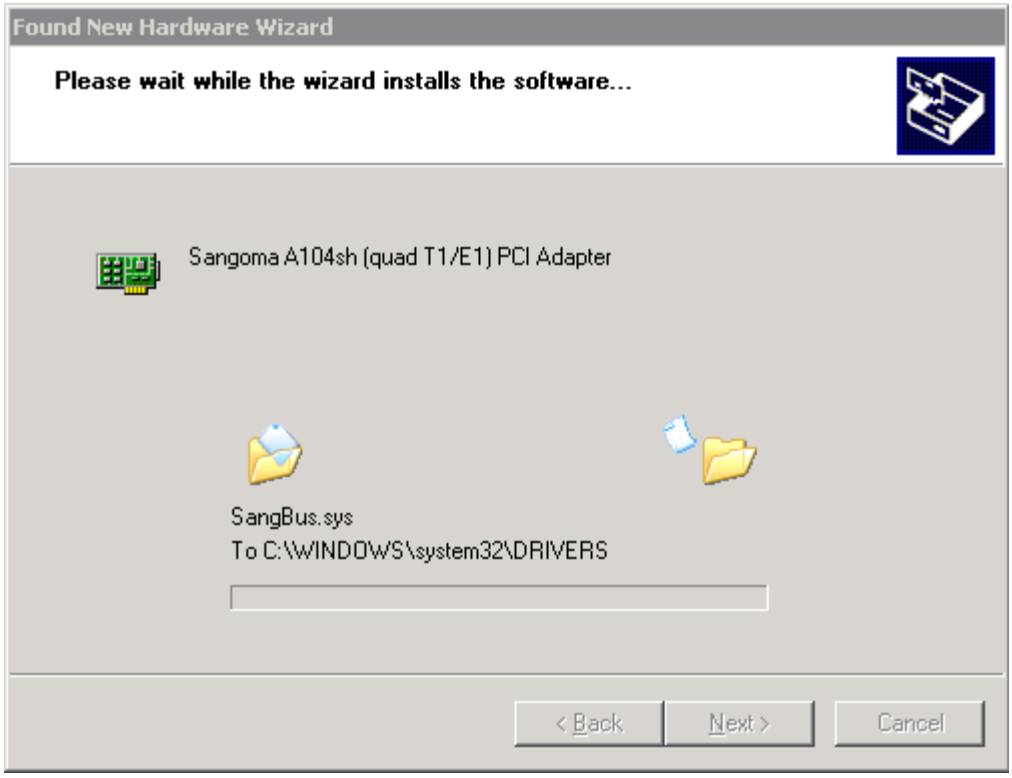
4. Click **Next**.

5. If a warning appears informing you that the software you are installing has not passed Windows logo testing, ignore the warning. Click **Continue Anyway**. Please refer to Appendix F: Modify the Microsoft Windows driver signing options to tell Windows not to ask for this confirmation.

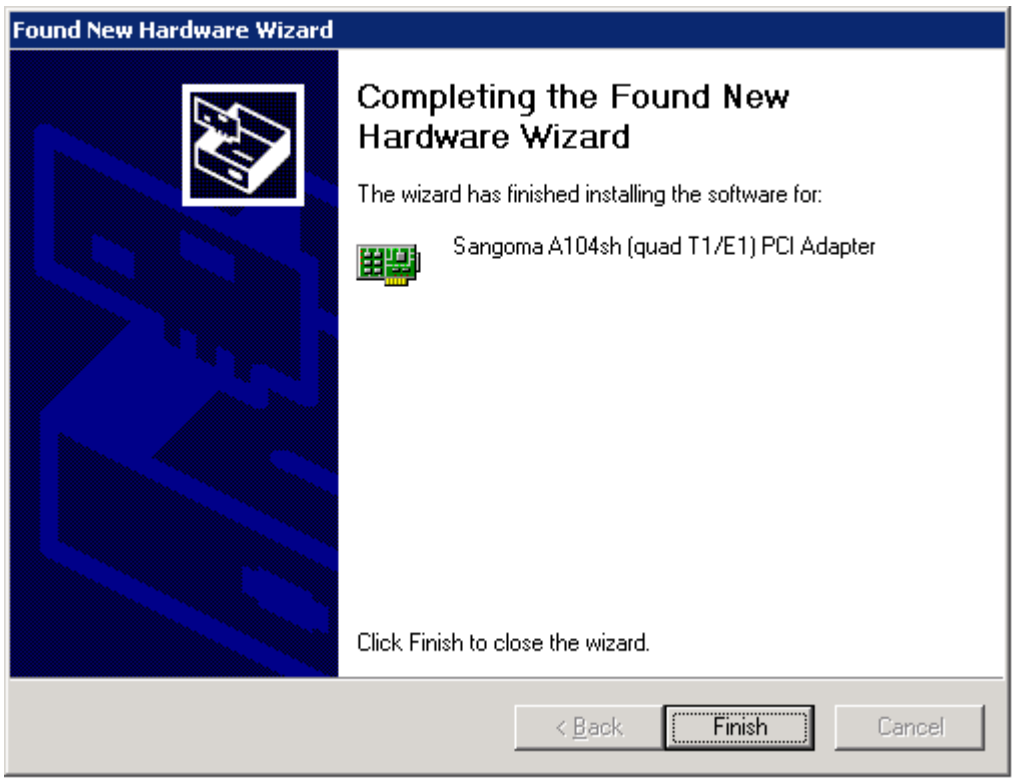


6. If the “Digital Signature Not Found” warning appears, ignore the warning and continue the installation.

Installation of the required network driver should proceed without interruption.



7. When the installation is complete, click **Finish**.



8. Proceed to install the drivers required for each of the board's interfaces. See Step 2: Installing the drivers for each of the interfaces on page 36.

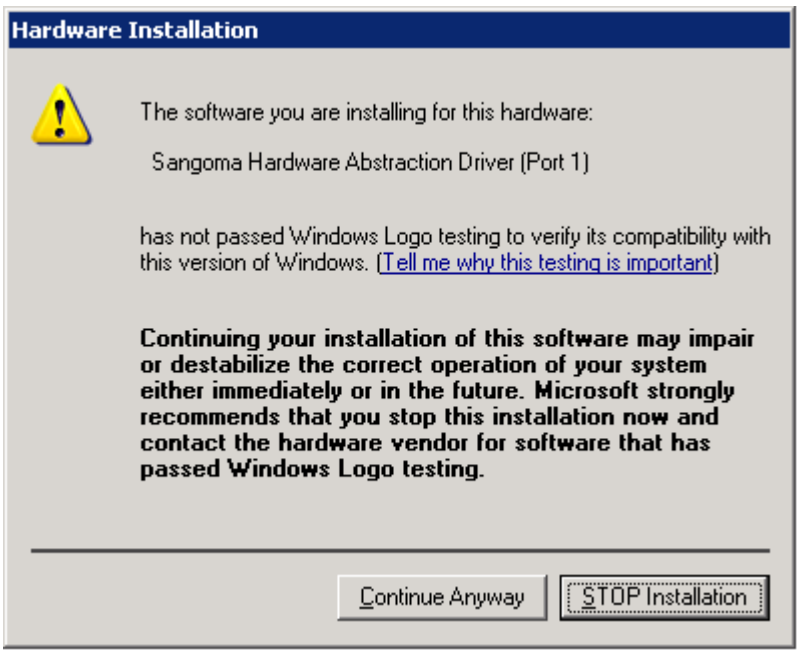
Step 2: Installing the drivers for each of the interfaces

Once the device driver for the Sangoma board is installed, your next step is to install the drivers for each of the required interfaces. This means that if you are using a Sangoma A104 quad, for example, you will need to install four separate device drivers.

The steps for installing the interfaces are very similar to those you just performed.

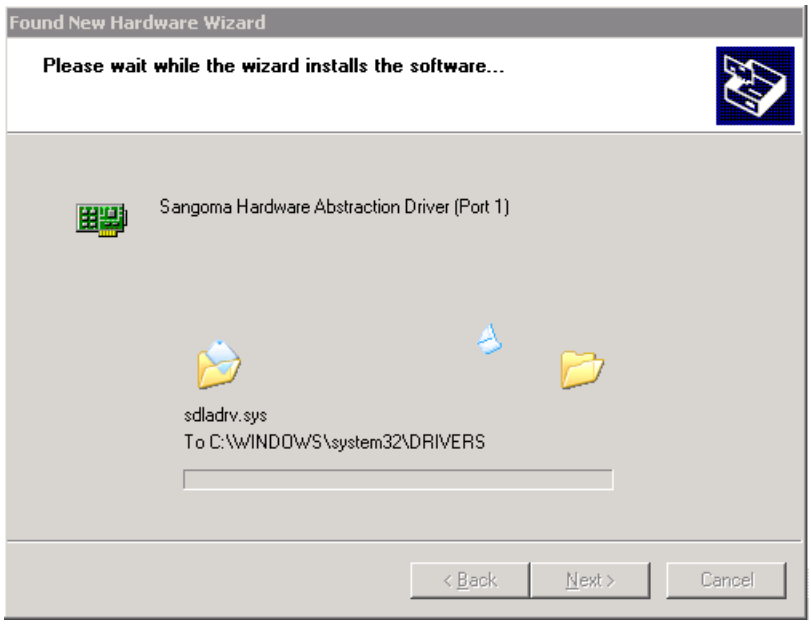
To install the required interfaces for the driver:

1. On the Welcome screen of the New Hardware Wizard, choose the “No, not this time” option to prevent Windows from searching for the required software through Windows Update.
2. Click **Next**.
3. Select the option “Install software automatically (Recommended)”,
4. Click **Next**.
5. If a warning appears informing you that the software you are installing has not passed Windows logo testing, ignore the warning. Click **Continue Anyway**.

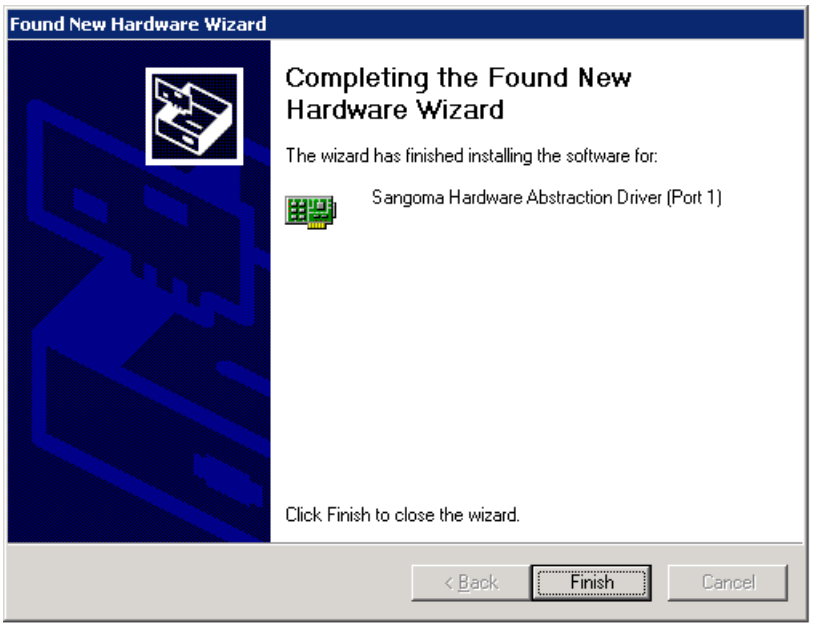


6. If the “Digital Signature Not Found” warning appears, ignore the warning and continue the installation.

   Installation of the required driver should proceed without interruption.



7. When the installation is complete, click **Finish**.



8. Continue the above steps (1 to 7) for each of the interfaces required for the Sangoma board.
9. Once you have completed installing the required drivers, restart your system. This step will ensure that the devices are available.
10. Proceed to device driver installation validation. See Step 3: Validating devices driver installation on page 38.

Step 3: Validating devices driver installation

1. Once the drivers are found and installed, and you have restarted the system, use the Windows Device Manager to confirm the successful installation of the driver for the PCI/PCI Express driver, as well as for the hardware abstraction drivers for each of the required interfaces.

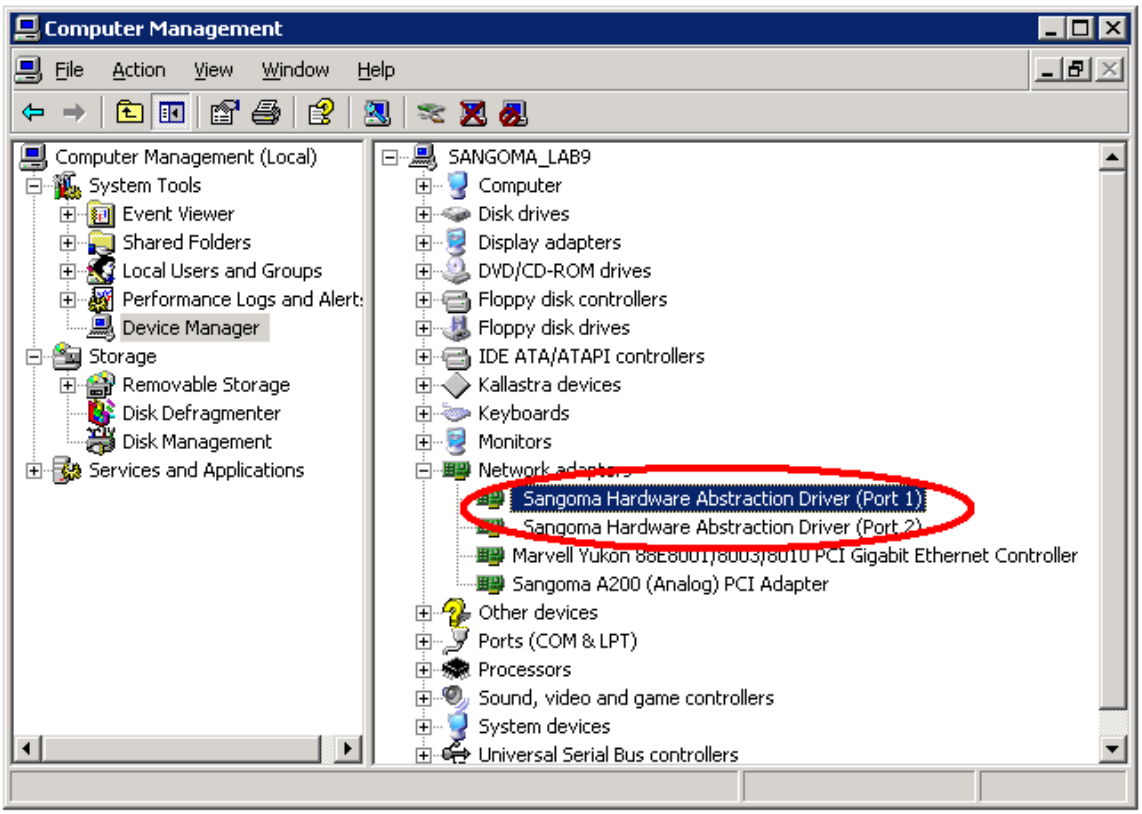
   To open the Device Manager, do the following:

   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 Device Manager from the Control Panel (double-click **System**, select the **Hardware** tab, and click **Device Manager**.)

   b) In the left pane, select "Device Manager" in the "System Tools" folder.

   c) If necessary, click the plus sign (+) beside "Network adapters" to expand the category.



In the example above, a driver for the Sangoma A104 board has been installed successfully, as well as the drivers for the four required interfaces.

If the Device Manager indicates an error, see What to do if the Device Manager indicates a device error on page 40.

What to do if the Device Manager indicates a device error

If the Device Manager indicates that one or more of the drivers was not installed properly, review this list for possible causes:

- Check if multiple versions of the drivers exist. If so, remove all previously installed Sangoma AFT device drivers and/or hardware from the system and re-install. **Note that the Gateway is compatible with Sangoma AFT device drivers v6.0.5.8; other versions are not compatible.**
- Check the Sangoma log file for errors. For information on locating the log file, see Sangoma device drivers log on page 87.
- Contact Sangoma Technologies Corporation support at support@sangoma.com.

Uninstalling the device drivers

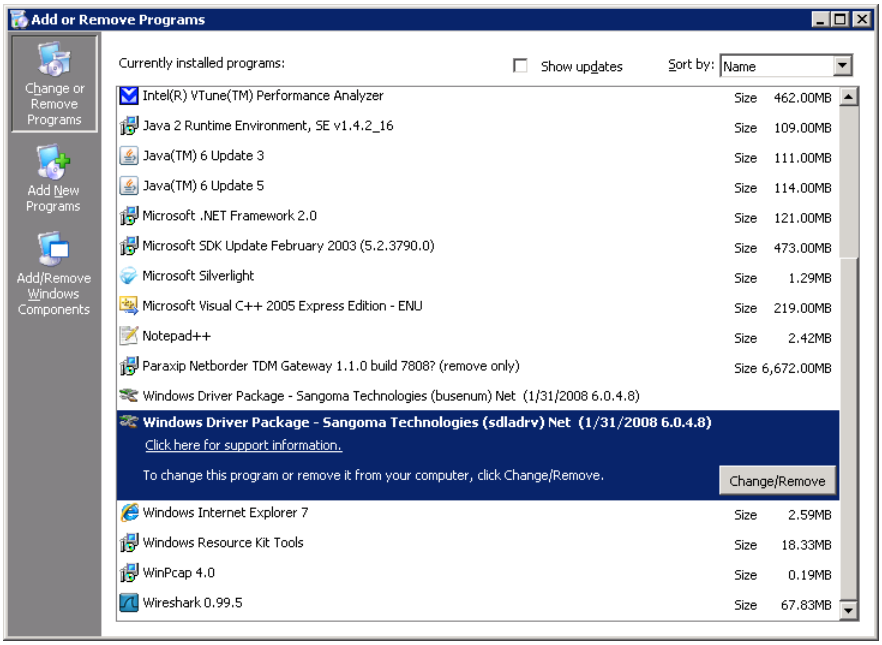
Use the Windows Device Manager to remove the drivers. The uninstall process is performed in reverse order to the installation process.

To uninstall the device drivers: Method A (recommended)

1. Stop the Gateway if it is running. See Managing the service on page 76.
2. To remove the Sangoma device drivers, from the **Start Menu**, select **Programs** > **Netborder Express Gateway** > **Remove Device Drivers**.
3. Follow the on screen instructions.
4. Remove the Sangoma board(s) from the chassis of the host system. See Removing the board from the chassis

To uninstall the device drivers: Method B

1. Stop the Gateway if it is running. See Managing the service on page 76.
2. To remove the Sangoma device drivers, from the **Start Menu**, select **Control Panel** > **Add or Remove Programs**.

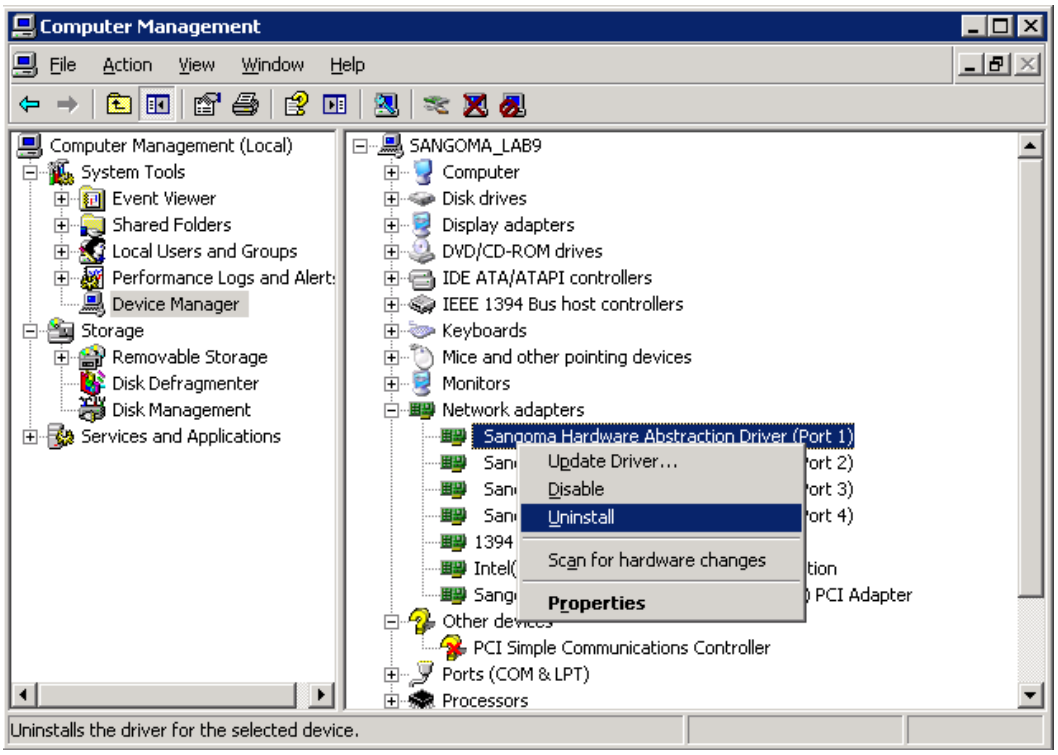


3. Select "**wanpipe**" driver, click on **Change/Remove** button.

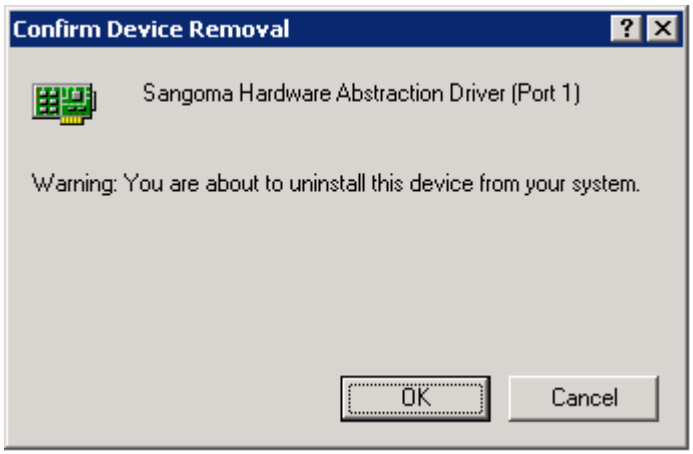
4. Follow the on screen instructions.
5. Select "**sprotocol**" driver, click on **Change/Remove** button.
6. Select "**sdladrv**" driver, click on **Change/Remove** button.
7. Select "**busenum**" driver, click on **Change/Remove** button.
8. Remove the Sangoma board(s) from the chassis of the host system. See Removing the board from the chassis

To uninstall the device drivers: Method C

1. Stop the Gateway if it is running. See Managing the service on page 76.
2. Using the Windows Device Manager, expand the network adapters list, select each of the following **in this order**, and either click the **Uninstall** icon, or right-click and select the **Uninstall** action:
   - Sangoma Hardware Abstraction Drivers for each of the interfaces (in the example below, four drivers must be uninstalled, one at a time).
   - Driver for the Sangoma board; for example, “Sangoma A104sh (quad T1/E1) PCI Adapter”.



3. Click **OK** to confirm removal of each of the drivers.



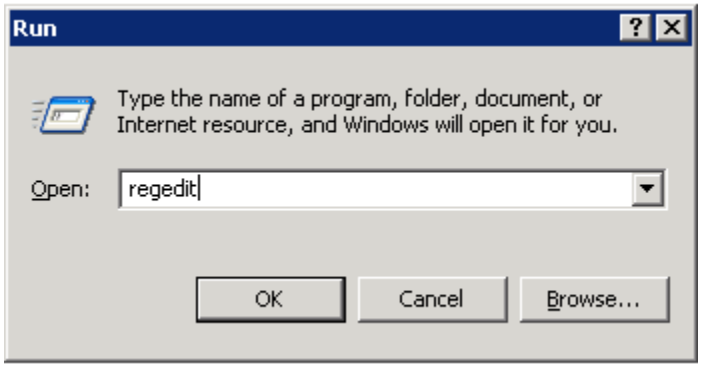
4. When all of the drivers have been uninstalled, use the System Registry to remove the “last_serial_number_range” value from the following file:

   - *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\busenum*

   Do the following:

   a) From the **Start Menu**, select **Run**.

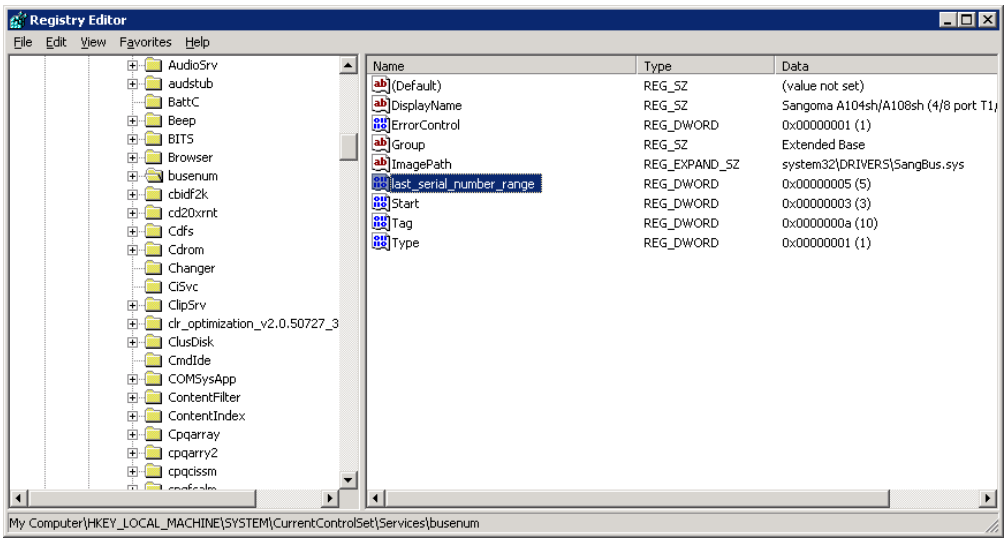
   b) Type “`regedit`” in the textbox, and click **OK**.



   c) In the Registry Editor, locate and select the following file:

      - *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\busenum*

   d) In the right pane, right-click the “last_serial_number_range” value and select **Delete** from the action menu (or simply left-click and press **Delete**).



5. Remove the Sangoma board(s) from the chassis of the host system. See below.

Removing the board from the chassis

Once the device drivers have been removed from the system, you may remove the Sangoma boards.

To remove a board from the chassis:

1. Shut down the system.
2. Unplug the board from PCI/PCI Express slot. For more information, refer to Inserting the board in the chassis on page 30.

Updating the device drivers

Use the Windows Device Manager to remove and reinstall the drivers. The update process is performed by removing the device drivers and installing the device drivers again.

To update the device drivers:

1. Stop the Gateway if it is running. See Managing the service on page 76.
2. To remove the Sangoma device drivers, from the **Start Menu**, select **Programs** > **Netborder Express Gateway** > **Remove Device Drivers**.

3. Follow on screen instructions until the device drivers removal process is completed
4. To install the Sangoma device drivers, from the **Start Menu**, select **Programs** > **Netborder Express Gateway** > **Install Device Drivers**.
5. Follow the on screen instructions.

Obtaining and installing the license file

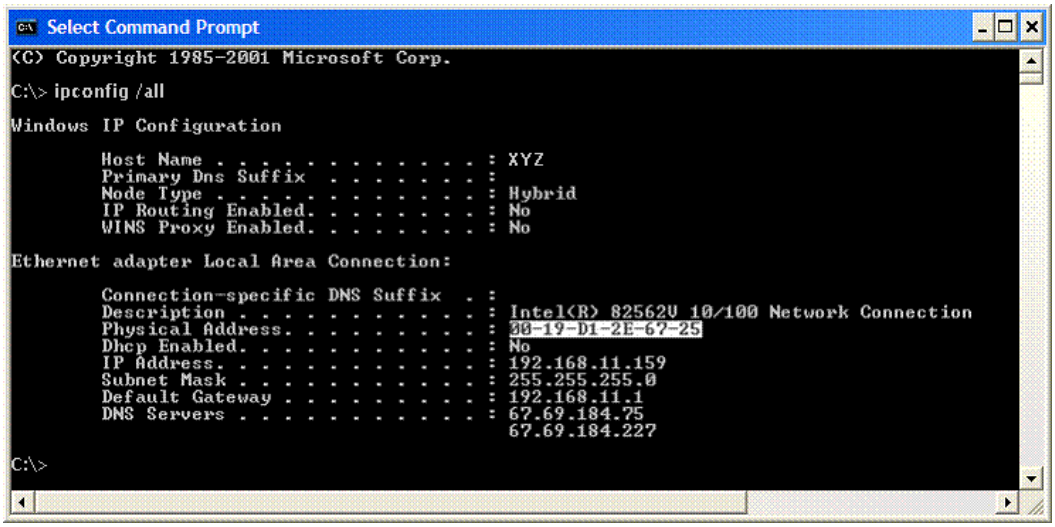
The Gateway is licensed on a per telephony port basis. The license is host locked.

To obtain a **full multi-port license**, simply obtain the *MAC (Media Access Control)* address of the server and contact our support organization at 1800 388 2475 or direct at +1 905 474 1990 or email at support@sangoma.com.

NOTE: The license provided via the Sangoma Technologies Corporation website is valid for 90 days from the date of download. If your license expires, download a new version.

To obtain the physical address of the Ethernet adapter:

1. Start a DOS command prompt and execute the following command:
   - `ipconfig /all`
2. Look for the *Physical Address* item. For example, it should look something like this:
   - `00-19-D1-2E-67-25`



Upon receiving your email providing the MAC address and required number of ports, Sangoma Technologies Corporation will send you new license files:

- *netborder-express-gateway-license.txt*
- *netborder-express-gateway-license.txt.sig*

To install the license file:

1. Once you have received the license files, copy the files (in effect, you will be replacing the temporary license) to the following directory:

   a) *[GATEWAY_HOME]\config*

      where *[GATEWAY_HOME]* is the root folder of the installation (for example, *C:\Program Files\Netborder\Express\Gateway\config*).

2. Restart the Gateway. From the **Start Menu**, select **Programs** > **Netborder Express Gateway** > **Stop Gateway** to stop the Gateway and select **Programs > Netborder Express Gateway > Start Gateway** to start the Gateway.

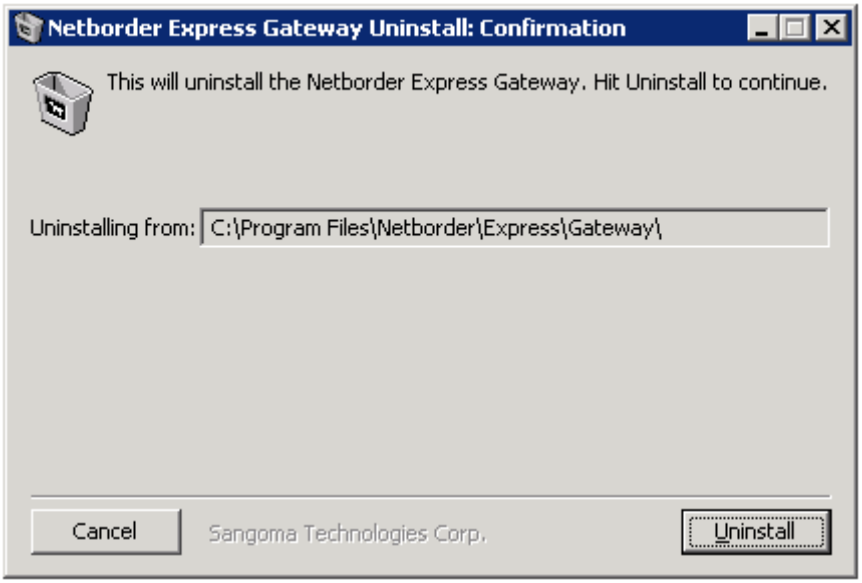
Uninstalling the software

The Sangoma device drivers must be uninstalled and removed from the host system before the Gateway software can be uninstalled.

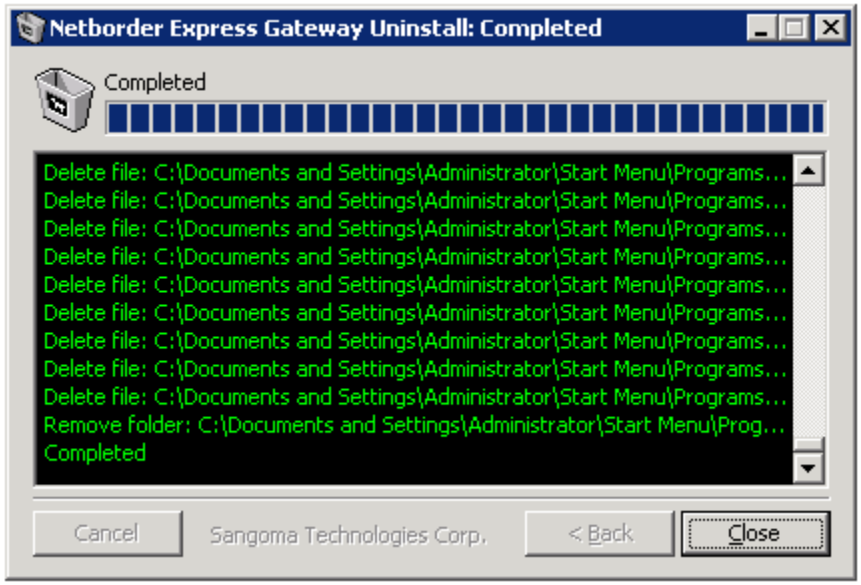
To uninstall the Netborder Express Gateway software:

1. Uninstall the Gateway software by launching the Gateway uninstaller from the Start Menu: **Start > Programs > Netborder Express Gateway > Uninstall**.

   The following window appears.



2. Click **Uninstall**.
3. Click **Yes** to remove the Sangoma device drivers when the uninstaller will ask if you want to install the device drivers.



The Gateway software is removed from your system.

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

Updating the software

The following procedure describes how to update the Netborder Express Gateway to a different version.

To update the Netborder Express Gateway software:

1. Save the license files found under "[GATEWAY_HOME]/config" into another directory.

- *netborder-express-gateway-license.txt*
- *netborder-express-gateway-license.txt.sig*

NOTE: Your licence files may not work for the new gateway version. The gateway will log an error when it will attempt to start if the license file are compatible. If you get this error please refer to section Obtaining and installing the license file to obtain the appropriate license files.

2. Uninstall the Gateway software as described in the section "Uninstalling the software" of the user guide for the version to be removed.
3. Delete any files and directories that be left behind by the uninstaller. In a Windows command windows (cmd.exe) enter the following command:

   a) del /S /Q [GATEWAY_HOME]

   where [GATEWAY_HOME] is the root folder of the installation (for example, C:\Program Files\Netborder\Express\Gateway\)
4. Install the new version of the Netborder Express Gateway software as described in section Installing the software

Validating the installation

Once you have installed the Gateway software and required Sangoma board, including device drivers for the board and each of the required interfaces, you should validate the successful installation and configuration.

To validate the installation, start the Gateway Web Interface. The Gateway's Web Interface is used to monitor and verify the status of the Gateway.

Starting the Web Interface

To start the Web-based monitoring tool:

1. From either Microsoft® Internet Explorer (version 6.0 and higher) or Mozilla Firefox™ (version 2.0.0 and higher), point to the following URL:

   a) *http://[hostname]:[port]/*

   where *[hostname]* is the name or IP address of the host running the Gateway service, and *[port]* is the port number (by default "*7782*").

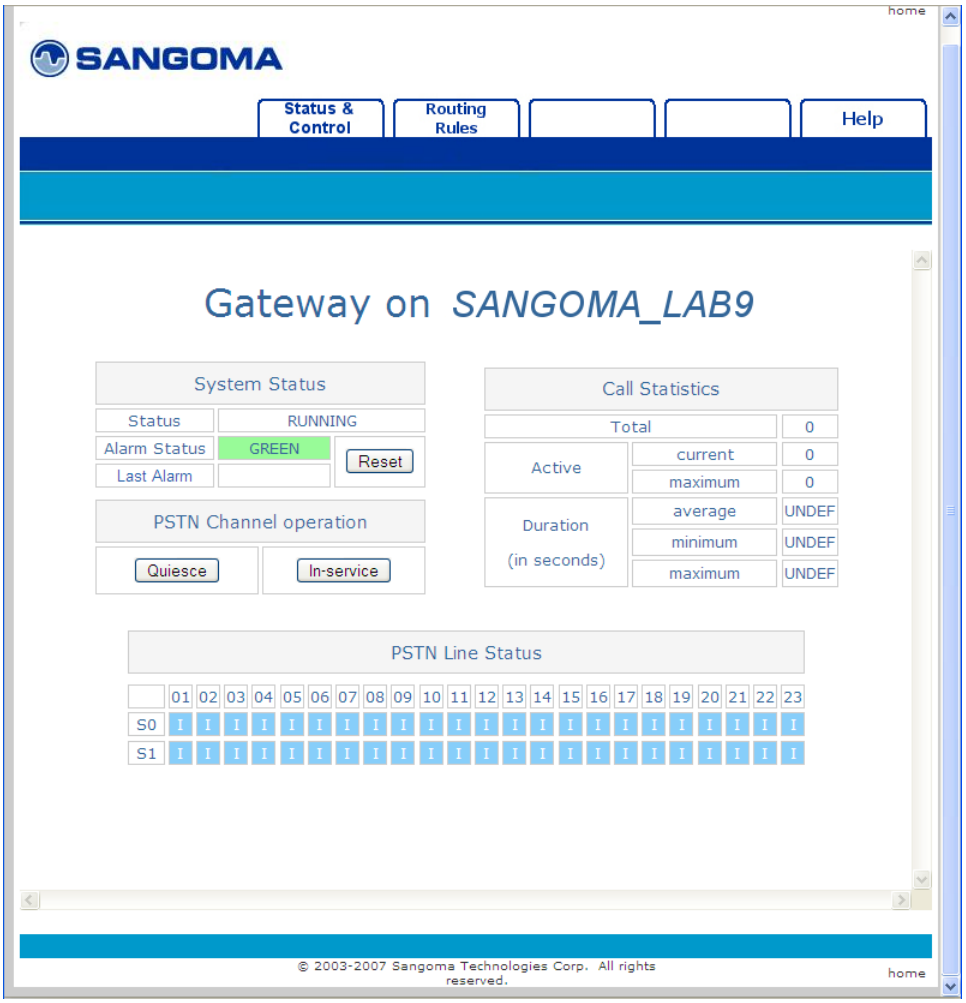
   For example, *http://XYZgateway:7782/*.

2. On the Status and Control page of the Web Interface (first page), verify the following:

   a) In the System Status area, the status of the Gateway is “green”.

   b) In the PSTN Line Status area, all configured channels are in “IDLE” state (each channel is blue and is accompanied by the letter **I** for **IDLE**).

   In the example that follows, note that two spans are functional, with 23 channels available to each span.



   You are now ready to place a call using the Gateway. Turn to Chapter 3: Getting started.

   However, if you encountered an error, such as an Alarm state, see What to do if the Gateway is in Alarmed state on page 53.

For more information on the Web Interface and its role as a monitoring tool, see Using the Gateway Web Interface on page 90.

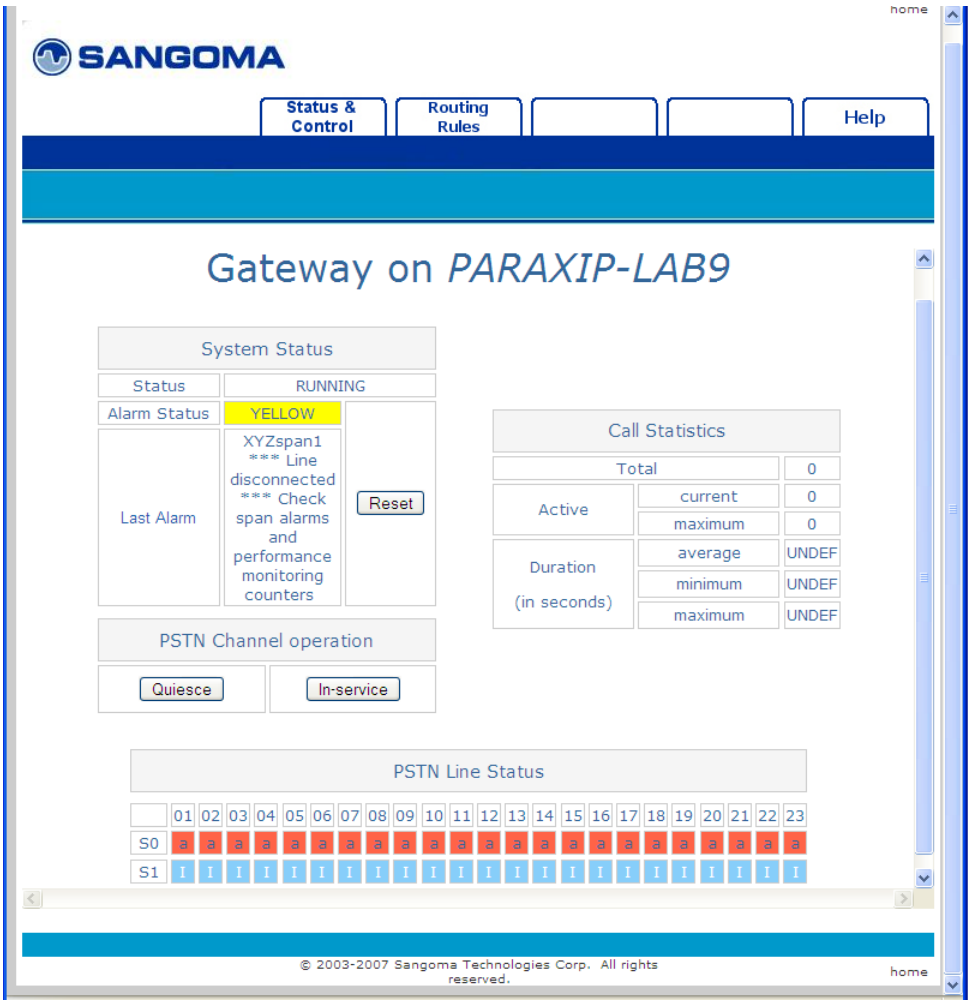
What to do if the Gateway is in Alarmed state

If the Web Interface indicates an “ALARMED” state, check the message received in the Last Alarm field. It should give you an indication of what has gone wrong.

If one of the channels is in an alarmed state, indicated in red in the PSTN Line Status area of the page, try one or more of the diagnostic steps below:

- Step 1: Ensure the Gateway Service is running.
- Step 2: Start the Windows Event Viewer to detect any errors with the Gateway software.
- Step 3: Check your connections.

For further assistance, see Chapter 9: Troubleshooting.



Step 1: Ensure the Gateway Service is running

It is possible that the Gateway has been stopped; for example, after a system restart.

Check that the Gateway Service is running using Services:

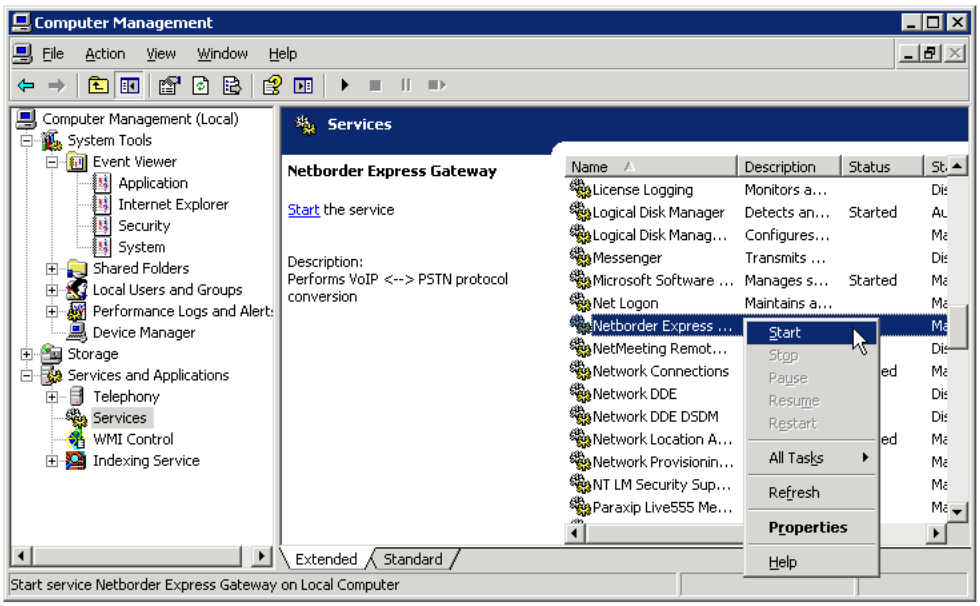
1. To open the Services list, do the following:
   a) Right-click **My Computer**, and click **Manage**.

   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.

2. In the Services list, right-click "Netborder Express Gateway" and select **Start** from the action menu.

   If the Gateway is already running, you will see its Status as "Started". The option "Start" will not be accessible from the right-click action menu.



Step 2: Start the Windows Event Viewer

The Windows Event Viewer provides detailed information about system events, including errors.

To start the Event Viewer:

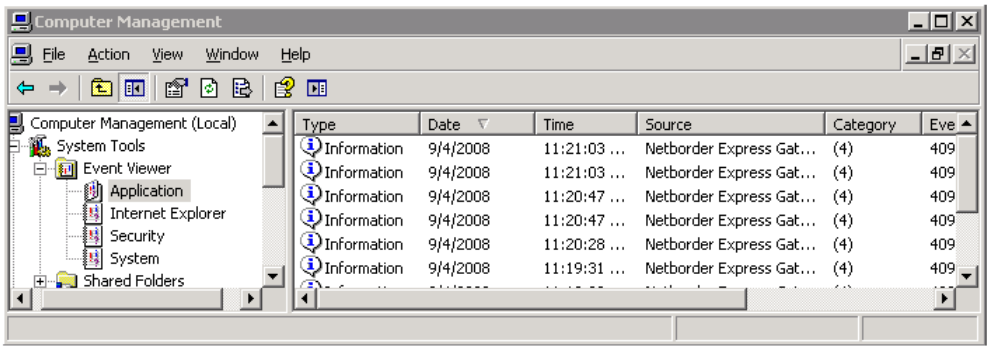
1. To open the Event Viewer, do the following:
   a) Right-click **My Computer**, and click **Manage**.

      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 open the Event Viewer from the Control Panel (double-click **Administrative Tools** and then double-click **Event Viewer**.)

   b) In the left pane, select "Event Viewer" under the "System Tools" folder.

   c) Expand the "Event Viewer" folder and select the "Application" folder.
2. In the right pane, verify that no warning or error has been detected with the Gateway.

To quickly find Gateway errors, click the “Source” event header to view events in ascending (alphabetical) order. Search through the list for any errors resulting from the “Netborder Express Gateway”.



Step 3: Check your connections

Ensure that all the IP connectivity to the system is functioning, and that the physical connectivity to the T1/E1 lines is correct. Check cable connections, and ensure you are using the correct cables.

Chapter 3: Getting started

Once you have installed and verified the Gateway service installation, you should place a call into the system to verify basic call control and audio control functions.

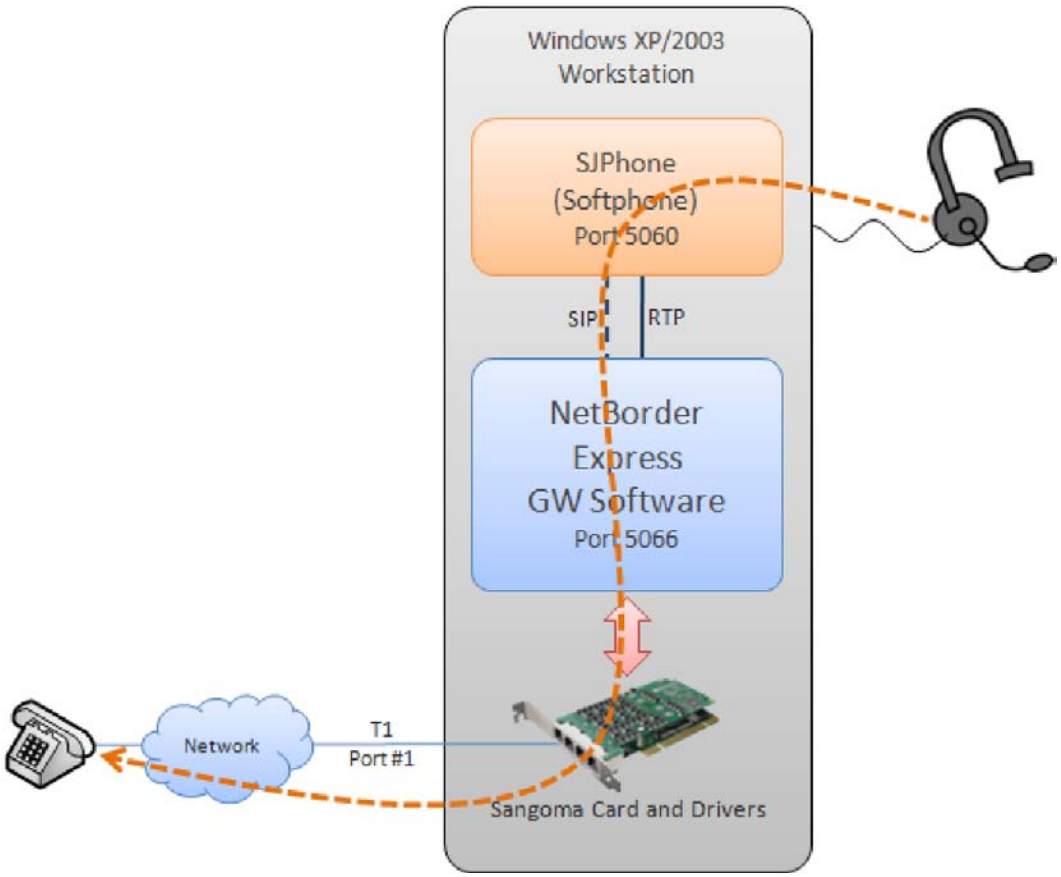
This chapter contains the following topics:

This chapter assumes that the Gateway software has been successfully installed on the target server that is, in turn, equipped with the appropriate telephony and media processing hardware and associated drivers. For installation procedures, please read Chapter 2: Installation.

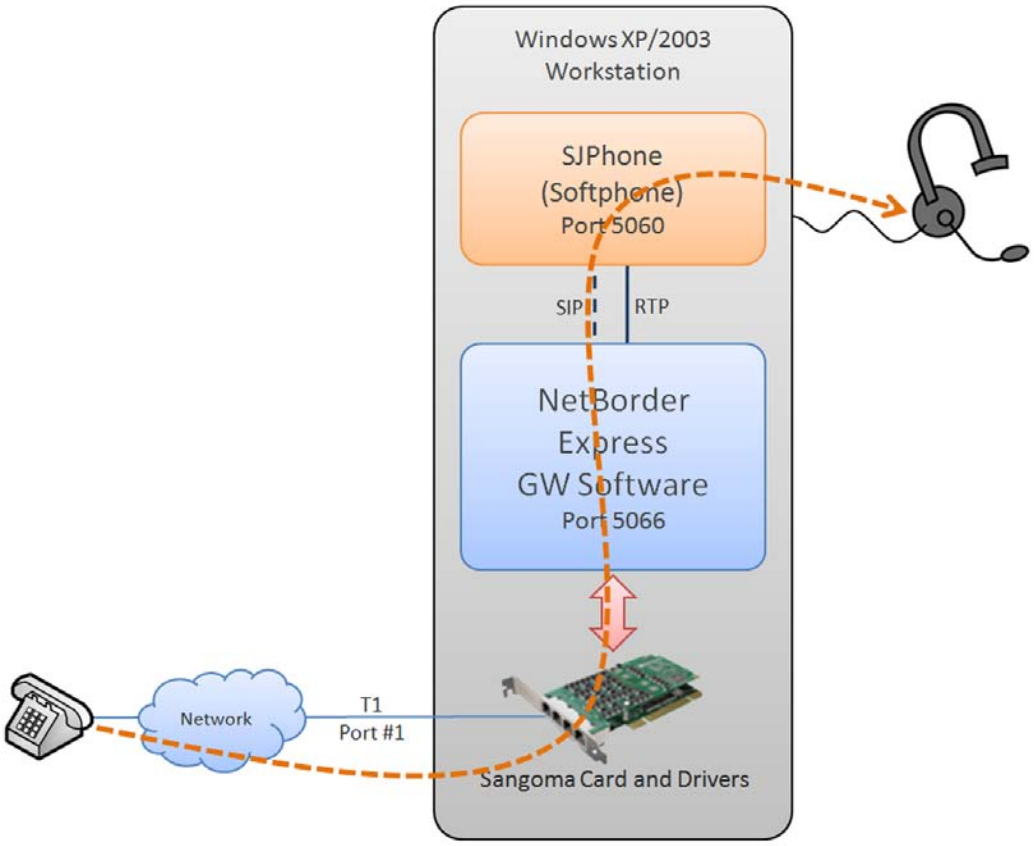
Basic call flow

The following diagrams illustrates the basic call flow for both SIP-initiated calls and PSTN-initiated calls using the Netborder Express Gateway.

SIP to PSTN call



PSTN to SIP call



Prerequisites to making SIP calls

Out of the box, the Gateway is able to connect with SIP application software responding to the default SIP URL specified during installation.

To make SIP calls, 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 SJ Labs' SJPhone.

NOTE: If you do not have a softphone installed on your PC, both of the products listed above are available for download at the URLs provided. Both Kapanga and SJ Labs provide on their websites technical specifications and information on configuring and using their application.

- Full-duplex sound card (which can record and play simultaneously; to test your card, try to run a recorder and a player at the same time).
- Speakers and microphone, or a headset.
- T1/E1 connection with a carrier or from a local network (remote end configured as "network").

Making a SIP to PSTN call

To verify that the SIP application (a SIP phone, for example) is successfully initiating outbound calls, place a call to a PSTN line.

To initiate an outbound call using a SIP phone:

1. Start the Gateway Web Interface. See Accessing the Web Interface on page 90.
2. Verify that the status of the Gateway is “green” and that all configured channels are in “IDLE” (**I**) state.
3. Start your softphone application.
4. Place a call to a PSTN phone number or extension using the softphone. Use dotted notation, as follows:

a) `[phone number or extension]@[IP address of Gateway]:[port]`

For example: 1026@192.168.11.207:5066

In the example above, an SJPhone is being used to call the telephone extension “1026”.

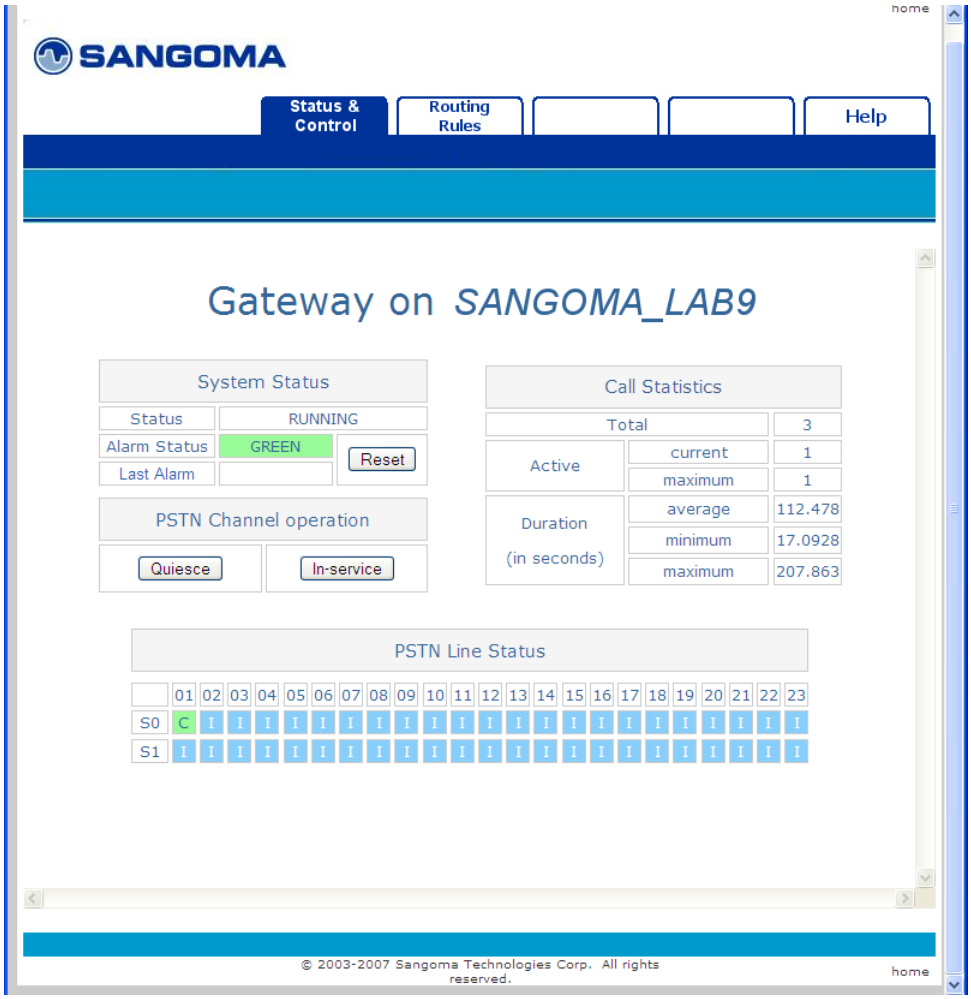
5. Connect the call.

   If you are using an SJPhone, press **Enter** or click the blue telephone icon in the lower right corner of the application.

6. While the call is being established, return to the Web Interface, and refresh the web browser to check that the call is going through properly.

   You should see in the associated telephony channel (the last channel, as specified by the default configuration) the following:

   - the letter **D** (for **D**ialing), to indicate that the channel is being used to initiate an outbound PSTN call
   - the letter **C** (for **C**onnected), to indicate that there is an active call on the telephony channel.



Note that in the above screenshots, the channels in the second span are in an alarmed state (indicated by the colour red and a lowercase “a”.)

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

   If you are using an SJPhone, click the red telephone icon in the lower right corner of the application.
10. As you are disconnecting the call, refresh the Web Interface to verify that the call is being disconnected. If you are quick enough, you will see the letter **H** (for **H**anging Up), indicating that the Gateway is in the process of disconnecting the call on that particular channel. Finally, the channel will return to **I** (IDLE) state.

For more information on the Web Interface and its role as a monitoring tool, see Using the Gateway Web Interface on page 90.

Making a PSTN to SIP call

To verify that a call is offered to the SIP application (a SIP phone, for example), place a call through the PSTN lines.

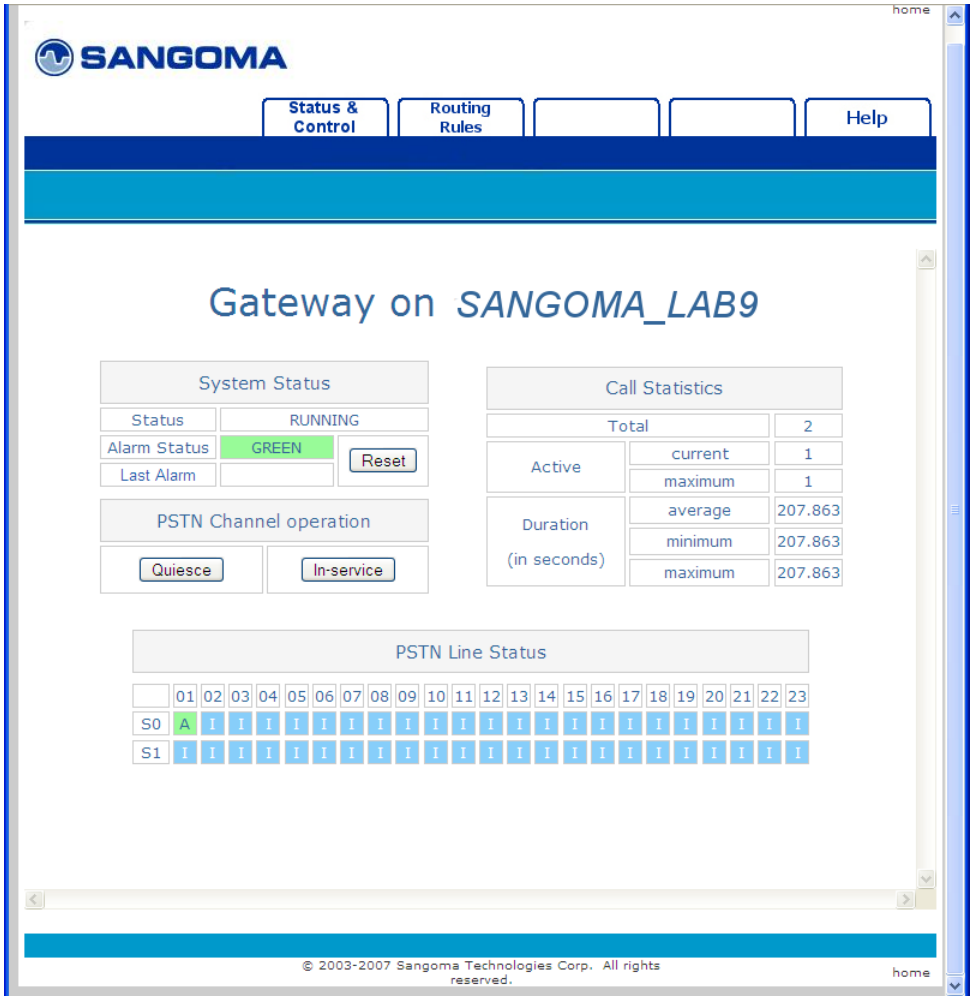
To initiate an incoming call to a SIP phone:

1. Start the Gateway Web Interface. See See Accessing the Web Interface on page 90.
2. Verify that the status of the Gateway is “green” and that all configured channels are in “IDLE” (**I**) state.
3. Start your softphone application.

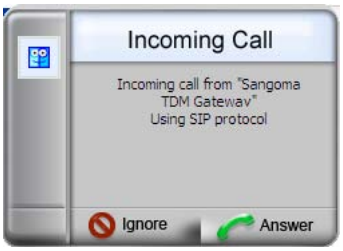
4. Place a call from a PSTN telephone to the SIP phone.
5. While the call is being established, return to the Web Interface, and refresh the web browser to check that the call is going through properly.

You should see in the associated telephony channel (the first channel, as specified by the default configuration) the following:

a) the letter **A** (for **A**nswering), to indicate that the channel has an active incoming PSTN call or request.



6. Answer the call using the SIP phone application.



If you are using an SJPhone, click the Answer button on the Incoming Call prompt.

7. Refresh the Web Interface to confirm that an active call is recognized on the telephony channel. You should see in the first channel **C** (for **C**onnected).
8. Interact with the application to verify that audio is coming through on both ends of the call.
9. When finished, hang up.
10. As you are disconnecting the call, refresh the Web Interface to verify that the call is being disconnected. If you are quick enough, you will see the letter **H** (for **H**anging Up), indicating that the Gateway is in the process of disconnecting the call on that particular channel. Finally, the channel will return to **I** (IDLE) state.

For more information on the Web Interface and its role as a monitoring tool, see Using the Gateway Web Interface on page 90.

Basic configuration changes

As stated earlier, out of the box the Gateway is able to connect with SIP application software responding to the default SIP URL specified during installation.

However, once you have verified basic call control and audio control functions by placing test calls, you may wish to make some basic configuration changes to, for example:

- run the Gateway as a standalone application (remotely as opposed to directly from the Gateway server), in which case you will need to edit the global configuration file (*gw.properties*)
- route inbound calls to a different SIP URI, in which case you will need to edit the routing rules file (*routing-rules.xml*).

See the procedures below.

Editing the global configuration file

Follow the steps of this procedure whenever you need to modify the *gw.properties* file (altering the appropriate parameter as required). For a complete list of parameters with a brief description, see Appendix B.

This procedure describes how to change the SIP port from the default, “5066”, to “5060”, the industry standard port number for SIP.

CAUTION: When using “5060” as the SIP port for the Gateway, it is important not to run any other applications that may cause conflict on that port (such as a SIP phone).

To run the Gateway remotely:

1. Open the global configuration file, *gw.properties*. From the **Start Menu**, select **Programs** > **Netborder Express Gateway** > **Configuration** > **Edit Global configuration file**.

   An editor will start allowing you to make changes to the *gw.properties* file.

2. Locate the following parameter: Netborder.*sip.userAgent.IPAddress*.

3. Change the value of this parameter to “5060”, as follows:

   a) `Netborder.sip.userAgent.IPAddress=INADDR_ANY:5060/udp, INADDR_ANY:5060/tcp`

   This parameter accepts as its value a comma-separated list of IP addresses and ports in the format: ip_addr1[:port1][/[udp|tcp]], ip_addr2[:port2][/[udp|tcp]]. Use “INADDR_ANY” as the IP address to listen on all IP interfaces on the host.

4. Save your changes.
5. Restart the Gateway. From the **Start Menu**, select **Programs** > **Netborder Express Gateway** > **Stop Gateway** to stop the Gateway and select **Programs > Netborder Express Gateway > Start Gateway** to start the Gateway.

Editing the routing rules file

Follow the steps of the procedure below whenever you need to change the *routing-rules.xml* file (altering the appropriate rule as required). For more information on routing rules and their function, including detailed information on how to modify them, see Chapter 7.

This procedure describes how to establish calls to a SIP proxy or another default SIP application living at a different URI.

To route inbound calls to a different SIP URI:

1. To make changes to the *routing-rules.xml* file, you have two options. Do **one** of the following:

   a) Using your favourite XML editor or any simple text editor (such as Notepad), locate and open the following file:

   - *[GATEWAY_HOME]\config\routing-rules.xml*

     where *[GATEWAY_HOME]* is the root folder of the installation (for example, *C:\Program Files\Netborder\Express\Gateway\config\routing-rules.xml*)

   - Using the Gateway Web Interface, go to the Routing Rules web page, and make your changes in the “Content of Routing Rules URL” text box. For details, refer to Using the Gateway Web Interface on page 90.

2. Locate the following routing rule: *default_sip_out*.

3. Change the “expr” attribute of the *sip.out.requestUri* element from “sip:localhost:5060” to the target SIP URI you wish to use. See the example provided.

```
<!-- SIP OUT TO sip:localhost:5060 -->
 <rule name="default_sip_out" outbound_interface="sip" qvalue="0.001">
  <condition param="transfer" expr="false"/>
  <condition param="pstn.in.channelName" expr=".*"/>
  <condition param="pstn.in.ani" expr="(.*)"/>
  <out_leg name="" media_type="sendrecv">
   <!-- To modify the target SIP destination, just change the value below
-->
   <param name="sip.out.requestUri" expr="sip:192.168.11.151:5090"/>
   <param name="sip.out.from.uri" expr="sip:%0@GW_HOST_IP:GW_SIP_PORT"/>
   <param name="sip.out.from.displayName" expr="Gateway"/>
   <param name="sip.out.transport" expr="udp"/>
  </out_leg>
 </rule>
```

4. Save your changes.
5. If you made changes to the *routing-rules.xml* file using an XML/text editor, you will need to restart the Gateway for your change to take effect. On the other hand, if you modified the file using the Web Interface, your changes will take effect on the next incoming call; there is no need to restart the Gateway.

Chapter 4: Operating the Gateway

This chapter describes how to operate the Gateway and how to monitor its performance using the Gateway's own Web Interface.

This chapter also provides information on the Web Service Management Interface, a programmable web service interface for customers who wish to either customize the management user interface, or integrate the Gateway's management capabilities into their existing framework. For details, see Using the Web Service Management Interface on page 95.

This chapter contains the following topics:

Setting service properties

In most cases, you will need separate services for your development and production environments. Isolating your development environment from your production applications will help you provide application stability and security to end users, while providing flexibility and freedom for development purposes.

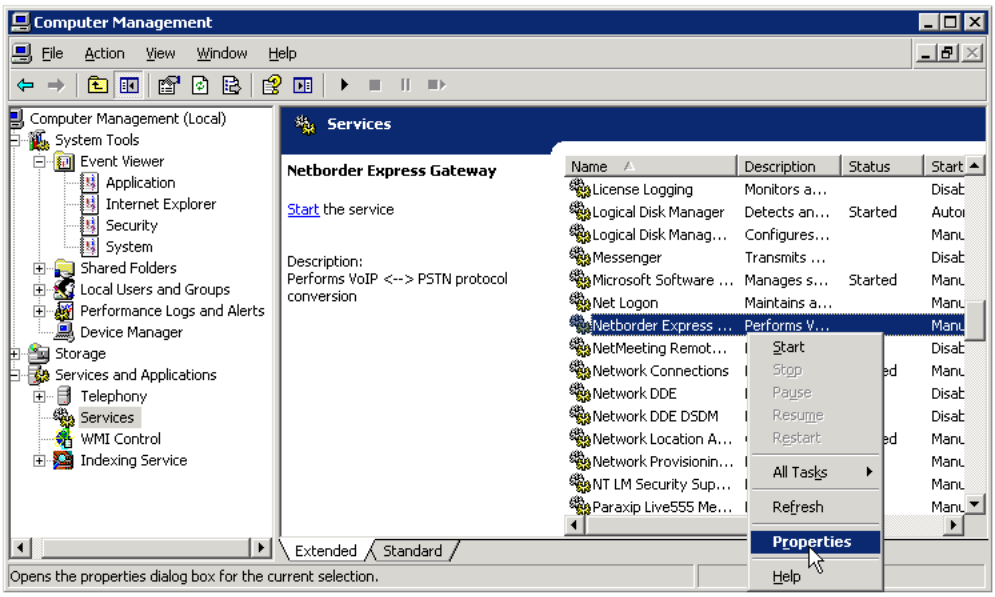
Modifications to the service are made using the Services list.

To configure the Gateway's service properties:

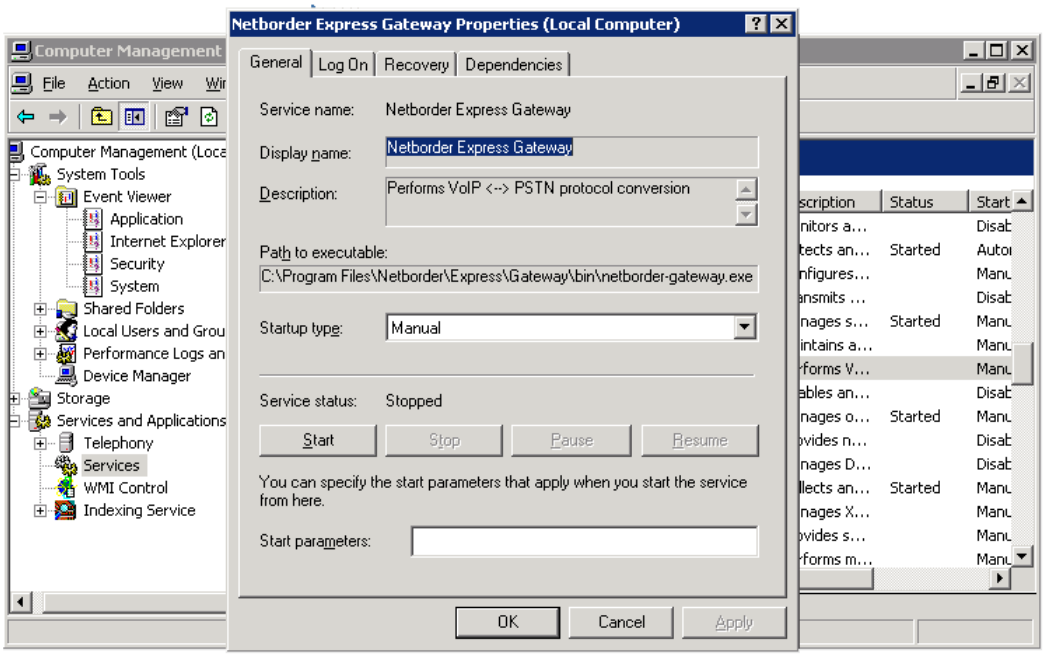
1. Open the Services list, as follows:
   a) Right-click **My Computer**, and click **Manage**.

   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.
2. In the Services list, right-click "Netborder Express Gateway" service, and select **Properties** from the context menu.



3. In the **General** tab, select your preferred **Startup type:**
   a) Developer mode: Select **Manual** (default). Service must be started manually (will not start automatically on system boot).
   b) Production systems: Select **Automatic**. Service will start automatically on system boot.



4. In the **Recovery** tab, select your preferred recovery procedure in the event of service failure:
   - Developer mode: Select **Take No Action** (default).
   - Production systems: Select **Restart the Service**. This setting will minimize downtime.
5. Click **OK**.

The Gateway allows you to maintain two custom logger configuration files, one for development and for production. For details, see Setting up logging on page 79.

Managing the service

Shortcuts have been provided in the Programs list to enable you to easily stop, start, and restart the Gateway. If the Gateway is in production mode, service will automatically start on system boot.

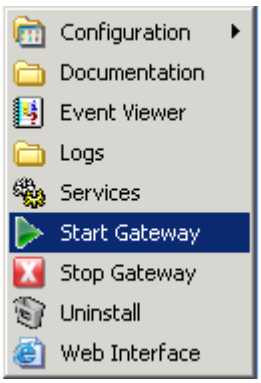
Occasionally you will need to stop, start and/or restart the Gateway manually. For example:

- Immediately after installation, you will want to restart the Gateway.
- After updating the configuration files (with the exception of the routing rules when edited using the Web Interface—see Editing the routing rules on page 93), you will want to restart the Gateway.
- **When in developer mode only**, upon system reboot or service failure, you will need to start the Gateway. **This is the default setting.** To change this setting, see Setting service properties on page 73.

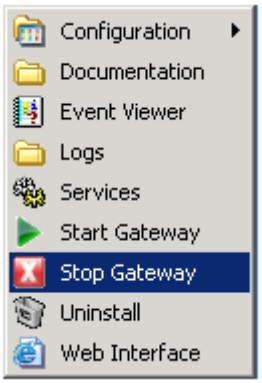
You have two options when it comes to starting, stopping or restarting the Gateway service. You may do so either from the Programs list or from the Services List. Both methods permit you to observe the status of the Gateway service directly at any time.

Programs list

To start the Gateway service from the Programs list, from the **Start Menu**, select **Programs** > **Netborder Express Gateway** > **Start Gateway**.



To stop the Gateway service from the Programs list, from the **Start Menu**, select **Programs** > **Netborder Express Gateway** > **Stop Gateway**.



Services list

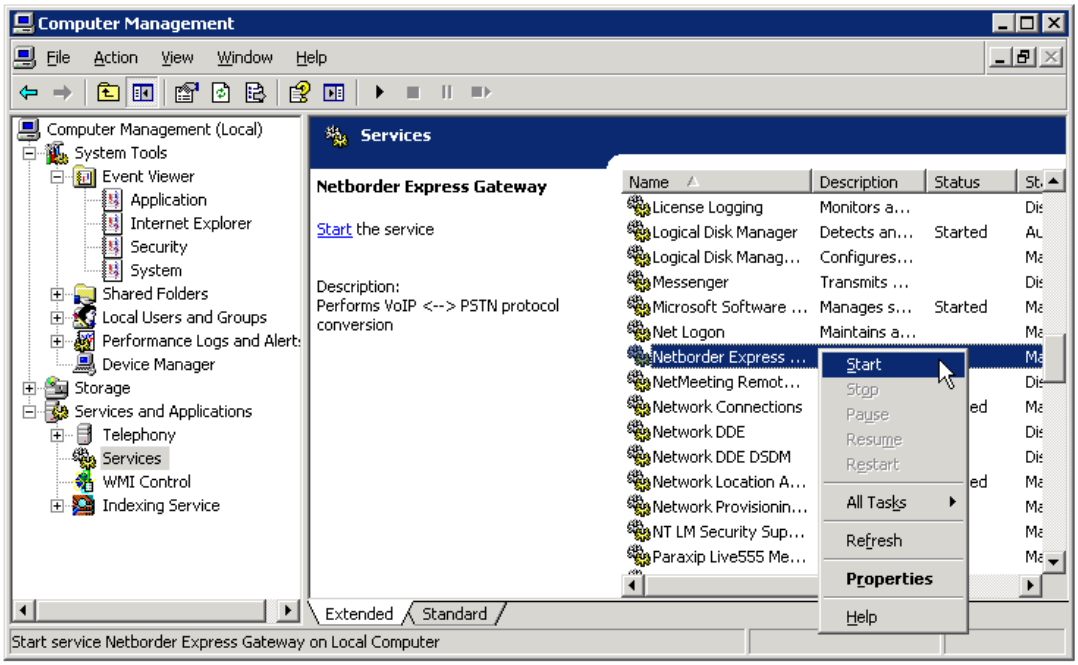
To check that the Gateway Service is running using Services:

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.
2. In the Services list, right-click "Netborder Express Gateway" and select **Start** from the action menu.

   If the Gateway is already running, you will see its Status as "Started". In this case, the option "Start" will not be accessible from the right-click action menu.



To manually stop or restart the Gateway service, follow the steps above, and select either **Stop** or **Restart**.

Setting up logging

The Gateway service includes a powerful logging framework to enable you to control the logging of events.

The Gateway has two logger configuration files:

- **dev-logger.properties:** Used in development mode.
- **prod-logger.properties:** Used in production mode.

Separate files allow you to maintain two custom logger configurations, one for development and for production. To switch between the two logger configurations, simply set the *Netborder.run.mode* parameter in the *gw.properties* file to '*production*' or '*development*'.

To change the *Netborder.run.mode* parameter:

1. Open the global configuration file, *gw.properties*. From the **Start Menu**, select **Programs** > **Netborder Express Gateway** > **Configuration > Edit Global configuration file**.

   An editor will start allowing you to make changes to the *gw.properties* file.

2. Locate the following parameter: Netborder*.run.mode*.
3. Change the logging configuration in one of two ways. Do **one** of the following:

   a) Change the value of the parameter to either "*production*" or "*development*". For example:

   ```
   Netborder.run.mode=production
   ```

   b) Remove the character '#' from the property you wish to enable. The '#' character denotes comments ignored by the software. To disable the previous logging configuration, precede it with the '#' character, so it is ignored by the system. For example:

   ```
   #Netborder.run.mode=development
   Netborder.run.mode=production
   ```

4. Save your changes.

5. Restart the Gateway. From the **Start Menu**, select **Programs** > **Netborder Express Gateway** > **Stop Gateway** to stop the Gateway and select **Programs** > **Netborder Express Gateway** > **Start Gateway** to start the Gateway.

Enabling/disabling logging

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.

Call log

By default the call log is enabled.

To disable the call log, do the following:

1. Open the following file with Notepad or the text editor of your choice:
   - *[GATEWAY_HOME]\config\gw.properties*

     where *[GATEWAY_HOME]* is the root folder of the installation (for example, *C:\Program Files\Netborder\Express\Gateway\config\gw.properties*).
2. Search for the following text string:
   - *log4cplus.appender.CALL_LOG_APPENDER=log4cplus::NullAppender*
3. Comment out the parameter by placing the character '#' directly in front of it, as follows:
   - *#log4cplus.appender.CALL_LOG_APPENDER=log4cplus::NullAppender*
4. Save your changes.

To enable the call log once again, ensure that the parameter above is not commented out. (Remove the character '#').

Logging PSTN alarms

If the Gateway is running in **development** mode, make sure the *dev-logger.properties* file contains the following parameter:

- *log4cplus.logger.Netborder.pstn.alarms=WARN*

If the Gateway is running in **production** mode, make sure the *prod-logger.properties* file contains the following parameter:

- *log4cplus.logger.Netborder.pstn.alarms=WARN*

If this parameter is preceded by the character '#', logging of PSTN alarms is disabled. Remove it to enable logging.

This parameter produces log events similar to the following:

```
007-10-01 17:39:14:547 [356] WARN - Netborder.pstn.alarms :
span[index=0,ID=Bell_span1]ALOS is ON
007-10-01 17:39:14:547 [356] WARN - Netborder.pstn.alarms :
span[index=0,ID=Bell_span1]LOS (Loss Of Signal) is ON
007-10-01 17:39:14:547 [356] WARN - Netborder.pstn.alarms :
span[index=0,ID=Bell_span1]RED alarm is ON
007-10-01 17:39:14:548 [356] WARN - Netborder.pstn.alarms :
span[index=0,ID=Bell_span1]OOF alarm is ON
007-10-01 17:39:15:547 [356] WARN - Netborder.pstn.alarms :
span[index=0,ID=Bell_span1]ALOS is OFF
007-10-01 17:39:15:547 [356] WARN - Netborder.pstn.alarms :
span[index=0,ID=Bell_span1]LOS (Loss Of Signal) is OFF
007-10-01 17:39:15:547 [356] WARN - Netborder.pstn.alarms :
span[index=0,ID=Bell_span1]RED alarm is OFF
007-10-01 17:39:15:548 [356] WARN - Netborder.pstn.alarms :
span[index=0,ID=Bell_span1]OOF alarm is OFF
```

Logging PSTN ISDN traces

If the Gateway is running in **development** mode, make sure the *dev-logger.properties* file contains the following parameter:

- *log4cplus.logger.Netborder.pstn.sangoma.isdn.message=DEBUG*

If the Gateway is running in **production** mode, make sure the *prod-logger.properties* file contains the following parameter:

- *log4cplus.logger.Netborder.pstn.sangoma.isdn.message=DEBUG*

If this parameter is preceded by the character '#', logging of PSTN ISDN traces is disabled. Remove it to enable logging.

This parameter produces log events similar to the following:

```
2007-12-11 13:15:04:172 [3748] DEBUG -
Netborder.pstn.sangoma.isdn.message.s3-Bell_span4-c1 :
call-id=1197396874-79875-41-5 : L2->PRI received DL_DA_IN on
pchan=0x0301 DISCONNECT
  Crv: 0x0001
  Codeset: 0
    CAUSE                    08 02 80 95
```

This example shows a “DISCONNECT” message received for channel “s3-Bell_span4-c1”. The message contains a CAUSE information element. The data contained in information element displayed beside the information element in hexadecimal format.

The channel is encoded as follow:

s<PSTN Interface index>-<PSTN Inferface ID>-c<B-Channel index>

For example, in the current example, the message was received for the 1st B-Channel on the "Bell_span4" PSTN interface.

Logging SIP messages

If the Gateway is running in **development** mode, make sure the *dev-logger.properties* file contains the following parameter:

- *log4cplus.logger.Netborder.sip.message=INFO*

If the Gateway is running in **production** mode, make sure the *prod-logger.properties* file contains the following parameter:

- *log4cplus.logger.Netborder.sip.message=INFO*

If this parameter is preceded by the character ‘*#*’, logging of SIP messages is disabled. Remove it to enable logging.

This parameter produces log events similar to the following:

```
2007-10-01 14:56:34:688 [3984] INFO - Netborder.sip.message : call-
id=1191264994-687500-41-0 RECEIVED SIP MESSAGE (REQUEST) via UDP from
192.168.11.159:5060 :
INVITE sip:1026@192.168.11.207:5066 SIP/2.0
Via: SIP/2.0/UDP
192.168.11.159;rport=5060;branch=z9hG4bKc0a80b9f00000078470142e70000276
f0000002e
From: "unknown" <sip:192.168.11.159>;tag=29eb4f5c77
To: <sip:1026@192.168.11.207:5066>;tag=ds-4823-475dba0c
Contact: <sip:192.168.11.159>
Call-ID: 9ADF7E76AEE54169B3946D13FB24D3E20xc0a80b9f
CSeq: 1 INVITE
Max-Forwards: 70
User-Agent: SJphone/1.65.377a (SJ Labs)
Content-Length: 371
Content-Type: application/sdp
Supported: replaces
Supported: norefersub
```

```
Supported: timer

v=0
o=- 3400253799 3400253799 IN IP4 192.168.11.159
s=SJphone
c=IN IP4 192.168.11.159
t=0 0
m=audio 49178 RTP/AVP 3 97 98 8 0 101
c=IN IP4 192.168.11.159
a=rtpmap:3 GSM/8000
a=rtpmap:97 iLBC/8000
a=rtpmap:98 iLBC/8000
a=fmtp:98 mode=20
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=setup:active
a=sendrecv
```

Logging RFC 2833 events

If the Gateway is running in **development** mode, make sure the *dev-logger.properties* file contains the following parameter:

- *log4cplus.Netborder.logger.media.jrtp =DEBUG*

If the Gateway is running in **production** mode, make sure the *prod-logger.properties* file contains the following parameter:

- *log4cplus.Netborder.logger.media.jrtp =DEBUG*

If this parameter is preceded by the character '#', logging of RFC 2833 events is disabled. Remove it to enable logging.

This parameter produces log events similar to the following:

```
2007-10-01 17:40:23:516 [2000] DEBUG - Netborder.media.jrtp.endpoint :
call-id=1191274818-531250-41-0 ept-id=0 New RFC2833 event --> start of
DTMF tone for digit='1'.
```

Viewing logs and events

This section describes briefly how to view logs for both the Gateway and Sangoma device drivers specific events. Again, detailed information on logging and logging configuration is contained in Appendix C.

Viewing Gateway 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 Gateway service to ensure that no warnings or errors occur, use the Window Events Viewer.

To use the Windows Event Viewer to view logging information:

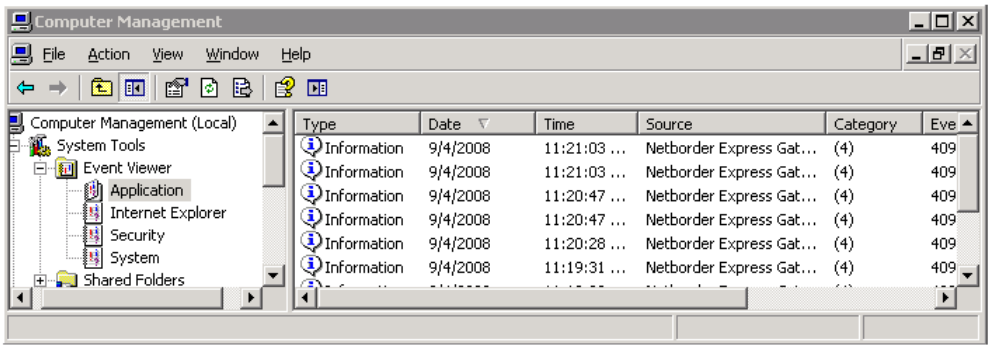
1. Open the Event Viewer, as follows:
   a) Right-click **My Computer**, and click **Manage**.

   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 open the Event Viewer from the Control Panel (double-click **Administrative Tools** and then double-click **Event Viewer**.)

   b) In the left pane, select “Event Viewer” under the “System Tools” folder.

   c) Expand the “Event Viewer” folder and select the “Application” folder.

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



3. Search through the list for information and/or any errors resulting from the “Netborder Express Gateway”.

Viewing Gateway logs of lower-level events

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

```
# Logger configuration

Netborder.run.mode=development

#Netborder.run.mode=production

Netborder.infra.Logger.PropertiesFile.development=C:\Program
Files\Netborder\Express\Gateway\config\dev-logger.properties

Netborder.infra.Logger.PropertiesFile.production=C:\Program
Files\Netborder\Express\Gateway\config\prod-logger.properties
```

In this example, to view logging events for production mode, you would open this file:

- *C:\Program Files\Netborder\Express\Gateway\config\prod-logger.properties*

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 224.

Sangoma device drivers log

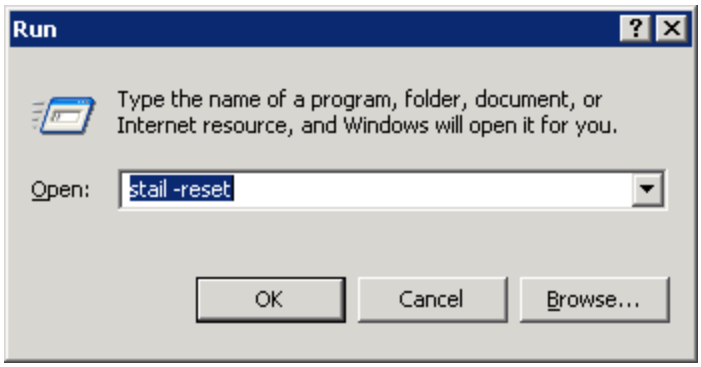
Sangoma device driver specific logging information is written to a file named *wanpipelog.txt*. This log file is located at:

- *%windir%\system32\drivers\wanpipelog.txt*

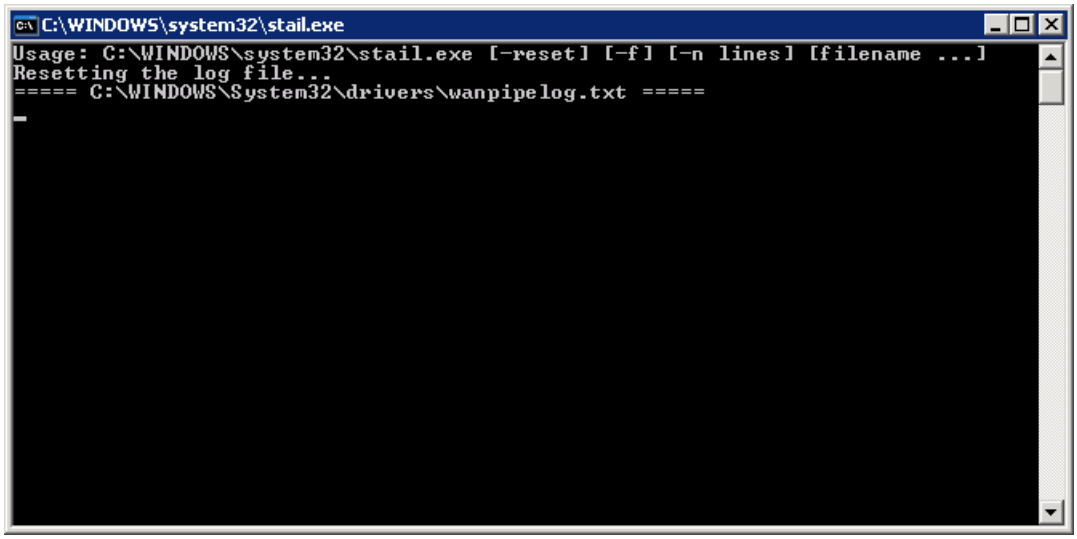
The size of the log file is limited to 500K and the device driver stop logging when the file size reach this limit.

To clear the log file use the Sangoma **stail.exe** utility application.

1. To start the **stail.exe** utility program. From the **Start Menu**, select **Programs** > **Run** and enter: ***stail.exe -reset*** in the Open text box of the Run dialog window.



2. To close the stail utility program. In the stail window, press **Ctrl + C**.



Interpreting PSTN alarms

This section describes common alarm and control signals that may appear during T1 and E1 operation. It also provides troubleshooting advice. (For more information on T1 and E1, see Step 3: Set the media type to T1 or E1 on page 106.)

The alarm signals have different colour designations and are used to indicate serious problems on the link. These alarm signals are defined as:

- Red Alarm
- Yellow Alarm
- Blue Alarm.

The gateway reports the alarms as log events. Please refer to Viewing logs and events for more details about log events.

Red Alarm

The red alarm is a local equipment alarm. It indicates that the incoming signal has been corrupted for a number of seconds. The red alarm shows up visually on the equipment that detects the failure. The equipment will then begin sending a yellow alarm as its outbound signal. This alarm is reported as "RED alarm" in the log.

Possible reasons for a red alarm include an invalid framing format. Check if the framing format configured on the port matches the framing format of the line. If it does not match, change the framing format on the controller to match the line. For more information on framing and how it relates to the PSTN configuration file, see Step 4: Set the framing and line encoding on page 109.

Yellow Alarm (Remote Alarm Indication Signal RAIS)

A yellow alarm indicates that the far-end equipment has encountered a problem with the signal it is receiving from the upstream equipment. This alarm is reported as "RAIS alarm" in the log.

Possible reasons include an invalid cable. Also, check the settings at the remote end to ensure that they match your port settings.

Blue Alarm (Alarm Indication Signal AIS)

A blue alarm indicates the total absence of an incoming signal. This alarm is reported as "AIS alarm" in the log.

Possible reasons include:

- The local or the remote end of the cable is not connected.
- The remote equipment is not powered up.

Using the Gateway Web Interface

The Gateway's Web Interface is used to monitor the Gateway's vital signs and to edit the routing rules. Sangoma Technologies Corporation has validated performance with both Microsoft® Internet Explorer (version 6.0 and higher) and Mozilla Firefox™ (version 2.0.0 and higher).

Accessing the Web Interface

To access the Gateway Web Interface, point your Internet browser to the URL of the Gateway's embedded web server. For example:

- *http://[hostname]:[port]/*

  where *[hostname]* is the name or IP address of the host running the Gateway service, and *[port]* is the port number (by default "*7782"*).

  For example: *http://XYZgateway:7782/*.

Or you can start the Gateway WEB Interface from the Programs list, from the **Start Menu.** Select **Programs** >**Netborder Express Gateway** > **Web Interface**.

NOTE: To modify the web server's port number, simply edit the file *[GATEWAY_HOME]/bin/shttpd.conf* and change the '*listen_port*' attribute.

Monitoring the Gateway's vital signs

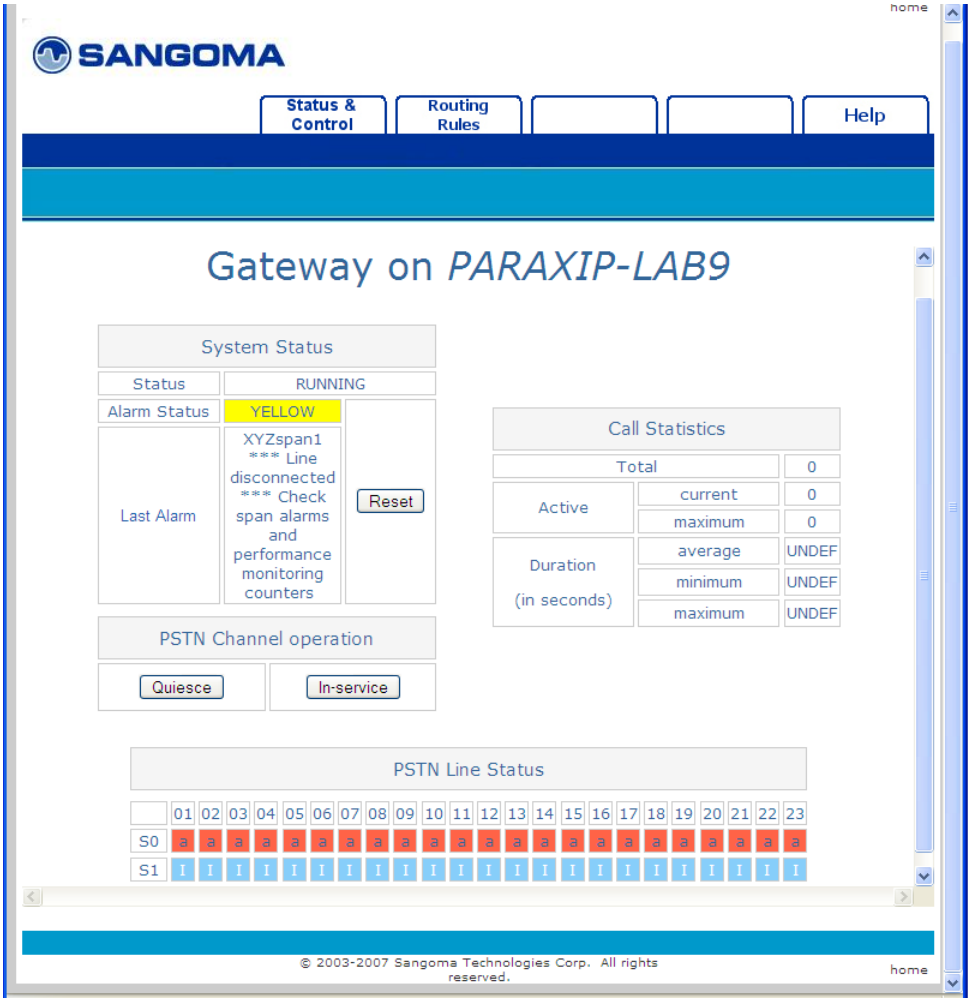
The first page displayed by the web browser is the **Status and Control** page. This page is used to monitor the Gateway's vital signs. Information provided includes:

- the Gateway alarm status – see System status on page 91
- call statistics – see Call statistics on page 92
- status of the telephony ports – see PSTN channel operation on page 92, and PSTN line status on page 92.

System status

If the Gateway alarm status is anything other than 'Green', then the last known alarm will be displayed, if any. Information related to the cause of the alarm is displayed in the Last Alarm field. To clear the alarm status, click the **Reset** button.

In the screenshot below, not only the Gateway has registered an alarm but also the Span 2 channels.



Call statistics

The following statistics are displayed:

- number of calls *since the Gateway service was started*
- number of currently active calls
- peak number of calls since the service was started
- minimum, maximum and average call durations in seconds.

PSTN channel operation

To *quiesce* the Gateway, which is to bring all of its ports out of service after the current calls are complete, click the **Quiesce** button. Click the **In-service** button to bring the Gateway ports back into service after a quiesce operation.

PSTN line status

Under the main display, the telephony channels are displayed, along with their status.

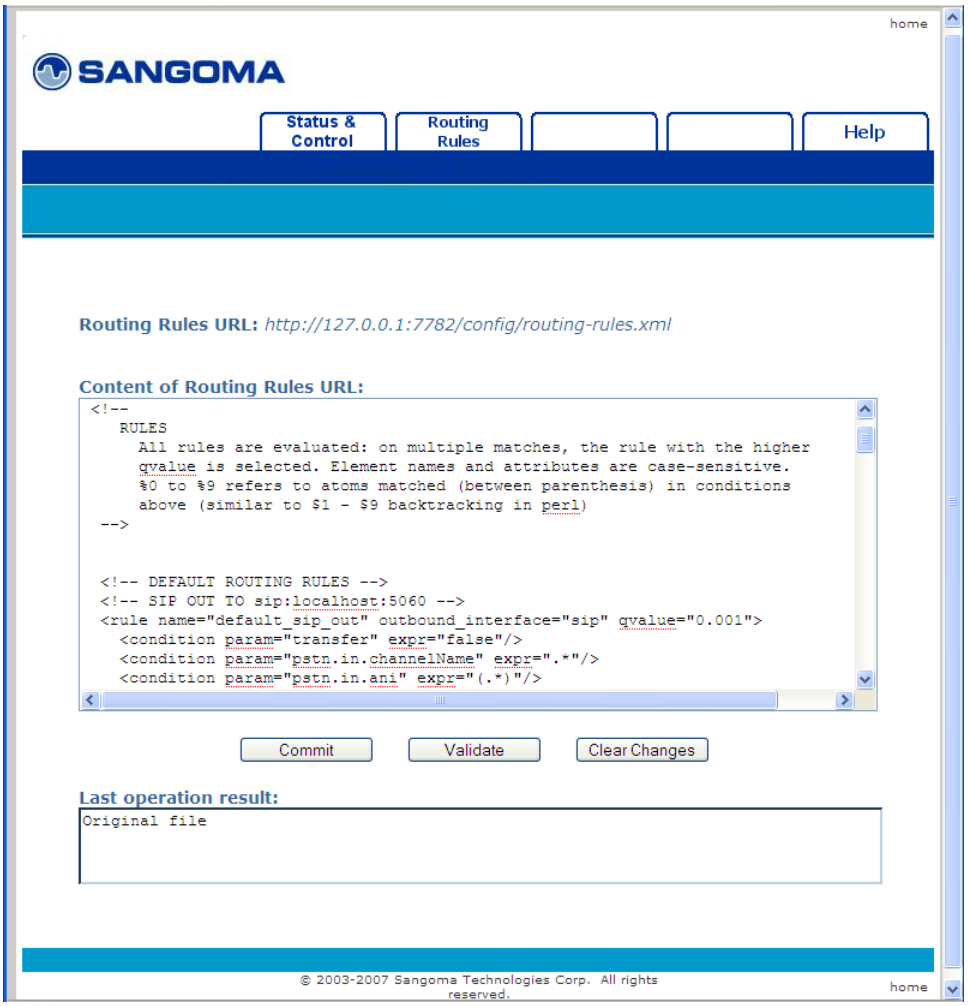
Telephony Channel Status		
Symbol & colour	***Status***	***Description***
a (red)	**ALARMED**	An alarm has been detected on the port. Verify the telephony connectivity into the Gateway.
I (blue)	**IDLE**	The port is ready to receive incoming PSTN calls and/or requests for outbound PSTN calls, depending on its configuration.
A (green)	**ANSWERING**	The port has an active incoming PSTN call and is in the process of setting up the associated VoIP leg so the call can be answered and connected.
D (green)	**DIALING**	The port is currently being used to initiate an outbound PSTN call.
C (green)	**CONNECTED**	There is an active call on the telephony port.
H (green)	**HANGING-UP**	The Gateway is in the process of disconnecting the call on that particular port.
G	**GOING-OUT-OF-SERVICE**	The port is being placed out of service.
O	**OUT-OF-SERVICE**	The port is out of service and cannot be used for either inbound or outbound calls.

Ports that are queued up for a quiesce operation will have an asterisk (*) beside the channel status.

NOTE: Use your browser's Refresh/Reload button to see more up-to-date statistics and status information.

Editing the routing rules

Routing rules can be edited directly from the Gateway Web Interface. Any changes made are dynamically updated, which means they will be used on the next incoming call. There is no need to restart the Gateway.



To edit the routing rules via the Web Interface:

1. Open the Gateway Web Interface. See Accessing the Web Interface on page 90.
2. Click the **Routing Rules** tab near the top of the web page to access the Routing Rules page.
3. Type your changes directly in the “Content of Routing Rules URL” text box.
4. Once your changes are complete, click the **Validate** button to validate the new rules.

   Validation results are reported in the “Last operation result” field near the bottom of the web page.

5. To save the new routing rules, click the **Commit** button.

   Once committed, the rules are permanently saved to disk and used for routing of the next incoming call. The result of the “Commit” operation is reported in the “Last operation result” field.

Accessing Help

From the Gateway Web Interface, just click on the **Help** tab to find links to the following support documents:

- Release Notes
- this guide
- a copy of your license
- the e-mail address of Sangoma Technologies Corporation Support.

Using the Web Service Management Interface

The Gateway offers a programmable web service interface for customers who wish to either customize the management user interface, or integrate the Gateway's management capabilities into their existing framework.

The interface uses the *Simple Object Access Protocol (SOAP)* as the messaging mechanism and *eXtensible Markup Language (XML)* as the data description format for communication between the Gateway's management agent and the customer's management client. The interface can be used to request information from the Gateway (for example, channel state, alarms, operational measurements), and to perform operations on the Gateway (such as change the logging level, execute maintenance operations such as quiesce, and reset alarms).

Those familiar with web services interfaces will find the relevant interface description in the *Web Services Description Language file (.wsdl),* located at *[GATEWAY_HOME]\doc\Service.wsdl* (where *[GATEWAY_HOME]* is the root folder of the installation).

Users can program either command line utilities using PERL or another "web services ready" scripting language, or HTML/PHP for graphical representations. Users who are not familiar with web services can refer to a number of books and online tutorials including the following:

- http://www.perfectxml.com/soaptutor.asp

For added convenience, Sangoma Technologies Corporation has packaged a Windows-based command line tool to invoke the main OAM commands without the necessity of writing any code. The executable can be found at `[`*GATEWAY_HOME]/bin/Netborder-oam-cmd.exe* (where *[GATWAY_HOME]* is the root folder of the installation).

From a console, for example, you would simply enter the following:

- Netborder`-oam-cmd.exe http://[gateway-host]:18086 [command]`

  where [`gateway-host]` is the host running the Gateway process and [`command]` is the web service command to execute.

Executing the tool without a command will invoke a usage example.

Chapter 5: PSTN

This chapter describes how to configure the parameters used to connect the Gateway to a traditional telephony TDM network such as the Public Switched Telephone Network (PSTN).

This chapter contains the following topics:

PSTN overview

The PSTN is a circuit-switched network. A dedicated circuit (also referred to as a channel) is established for the duration of a transmission, such as a telephone call. This contrasts with packet switching networks (such as the Internet), in which messages are divided into small segments called packets and each packet is sent individually. The PSTN is almost entirely digital, even though most subscribers are connected via analog circuits, and it now includes mobile phones in addition to fixed-line phones. Digital connections to the PSTN have been made available to end users through services such as ISDN (Integrated Services Digital Network).

In order to place a voice call over the PSTN, the telephony equipment sends a request to the PSTN to establish a circuit between the caller and the callee. This request is performed by means of call control protocols such as ISDN-PRI (Primary Rate Interface) or *CAS (Channel Associated Signaling)*. Once the circuit is established, the telephony equipment is able to exchange the voice over that circuit.

The Gateway uses the telephony boards from Sangoma to connect the Gateway to the PSTN. These boards permit the PSTN interfaces to switch voice calls over the PSTN.

To successfully transmit voice calls over the PSTN configuration, you must configure the following:

- PSTN interface parameters, such as the number of interfaces and their type (T1/E1) – see Configuring the interfaces on page 99
- Media parameters, such as echo cancellation and media encoding (PCM ulaw or PCM A-Law) – see Enabling/disabling echo cancellation on page 114
- ISDN signaling parameters, such as the protocol type (ISDN-PRI, CAS) and switch variant – see Configuring ISDN signaling on page 118
- Resources used by the Gateway in association with routing rules, such as for inbound and/or outbound calls – see Creating and configuring a resources group on page 135.

Accessing the PSTN configuration file

To configure the required PSTN interfaces, you will make changes to the PSTN configuration XML file. For a sample of a complete PSTN configuration file, see Appendix E.

The PSTN configuration XML file is located here:

- *[GATEWAY_HOME]\config\pstn-config.xml*

  where *[GATEWAY_HOME]* is the root folder of the installation (for example, C:\Program Files\Netborder\Express\Gateway*config\pstn-config.xml*).

The PSTN configuration XML schema is located here:

- *[GATEWAY_HOME]\config\pstn-config.rng*

Changes to the files may be made by any text editor or XML editor.

Configuring the interfaces

To configure the PSTN interface, your first step is to determine the number of interfaces that will be required. To provision the Gateway with the correct number of interfaces, you may need to add (install) or remove (uninstall) Sangoma telephony boards in the Gateway, as required. For more information, see Installing/uninstalling the Sangoma board on page 30.

Step 1: Create unique sangoma identifier(s)

Sangoma interface identifiers are used to refer to the interface in log events and various configuration parameters. **Sangoma interface identifiers must be unique.**

To create sangoma interface elements:

1. For each Sangoma interface, create a unique "sangoma" interface element in the PSTN configuration file. Each identifier is specified using the "*ID*" attribute of the "*sangoma*" element.

CAUTION: It is not recommended to declare fewer interfaces in the PSTN configuration file than the actual number of Sangoma interfaces installed in the system.

In the example below, two Sangoma PSTN interfaces have been declared, "XYZ`span1`" and "XYZ`span2`", as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- RelaxNG shema=pstn-config.rng -->
<pstnConfig version="1.0">

 <!-- ... -->

 <!-- All PSTN interfaces -->
 <interfaces>
  <sangoma id="XYZspan1"
      wanpipe="1"
      mediaType="T1"
      framing="ESF"
      lineEncoding="B8ZS"
      LBO="0dB"
      clocking="TERMINAL"
```

```
    loopbackMode="DISABLED">
   <BChannels defaultPcmLaw="PCMU"
          voicePacketLengthInMs="20">
   <VoiceQualityEnhancement>
    <EchoCancellation mode="NORMAL"
                comfortNoiseMode="NORMAL"
                tailDisplacementInMs="0"
                doubleTalkBehavior="OPTIMAL"/>
    <AcousticEchoCancellation mode="NORMAL"/>
    <LevelControl direction="ALL"
              mode="AUTOMATIC"
              targetLevelIndBM0="-20"/>
    <BackgroundNoiseAttenuation direction="FROM_PSTN"
                        mode="ENABLED"
                        attenuationIndB="-18" />
    <DTMFRemoval direction="FROM_PSTN"
             mode="DISABLED"/>
   </VoiceQualityEnhancement>
  </BChannels>
  <DChannel channelIndex="24"
        mtuSize="2048"/>
</sangoma>
<sangoma id="XYZspan2"
    wanpipe="2"
    mediaType="T1"
    framing="ESF"
    lineEncoding="B8ZS"
    LBO="0dB"
    clocking="TERMINAL"
    loopbackMode="DISABLED">
   <BChannels defaultPcmLaw="PCMU"
          voicePacketLengthInMs="20">
   <VoiceQualityEnhancement>
    <EchoCancellation mode="NORMAL"
                comfortNoiseMode="NORMAL"
                tailDisplacementInMs="0"
                doubleTalkBehavior="OPTIMAL"/>
    <AcousticEchoCancellation mode="NORMAL"/>
    <LevelControl direction="ALL"
              mode="AUTOMATIC"
              targetLevelIndBM0="-20"/>
    <BackgroundNoiseAttenuation direction="FROM_PSTN"
                        mode="ENABLED"
                        attenuationIndB="-18" />
    <DTMFRemoval direction="FROM_PSTN"
             mode="DISABLED"/>
```

```
      </VoiceQualityEnhancement>
    </BChannels>
    <DChannel channelIndex="24"
            mtuSize="2048"/>
  </sangoma>
 </interfaces>

 <!-- ... -->

</pstnConfig>
```

Step 2: Associate each interface with a wanpipe number

Wanpipe numbers are assigned when the Sangoma board is installed in the chassis. Usually the first interface of the first board installed in the chassis is assigned to wanpipe=1, the second interface to wanpipe=2, and so on. Use the Windows Device Manager to determine the wanpipe number associated with each sangoma interface.

To associate each interface with a "wanpipe" attribute:

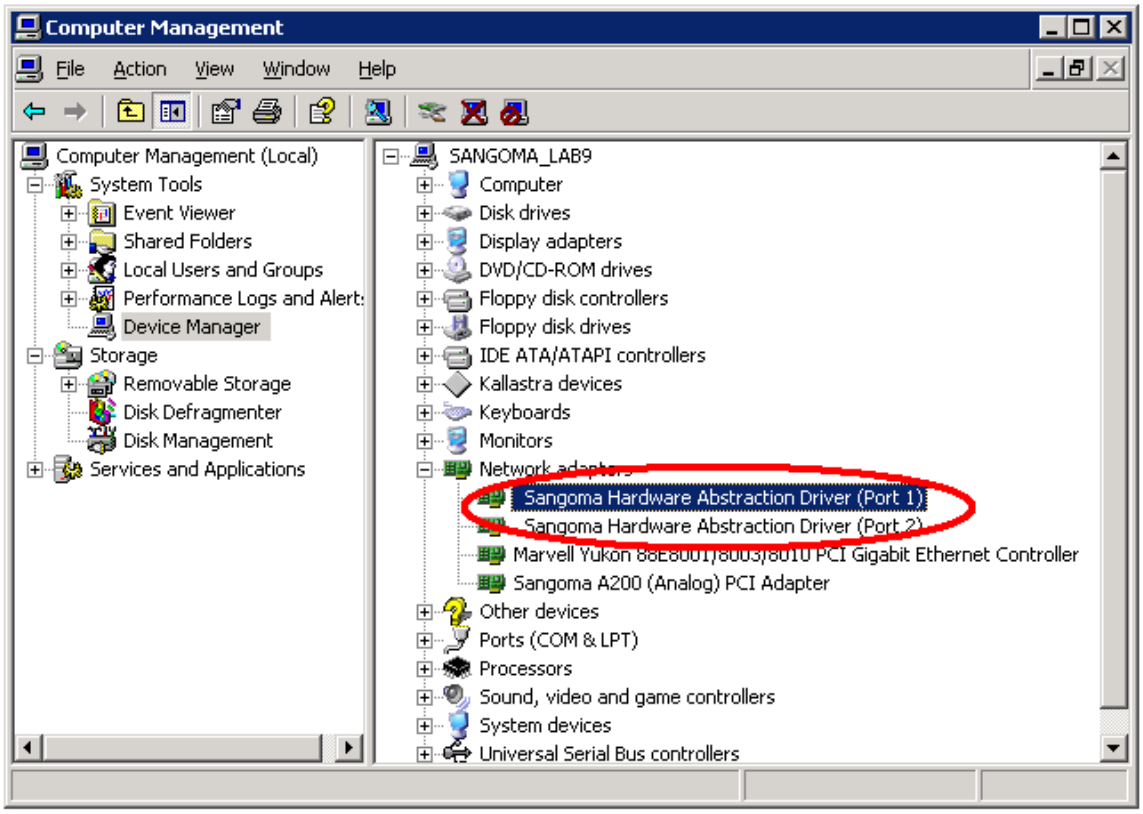
1. Open the Windows Device Manager, as follows:
   a) Right-click **My Computer**, and click **Manage**.

   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 Device Manager from the Control Panel (double-click **System**, select the **Hardware** tab, and click **Device Manager**.)

   b) In the left pane, select "Device Manager" in the "System Tools" folder.

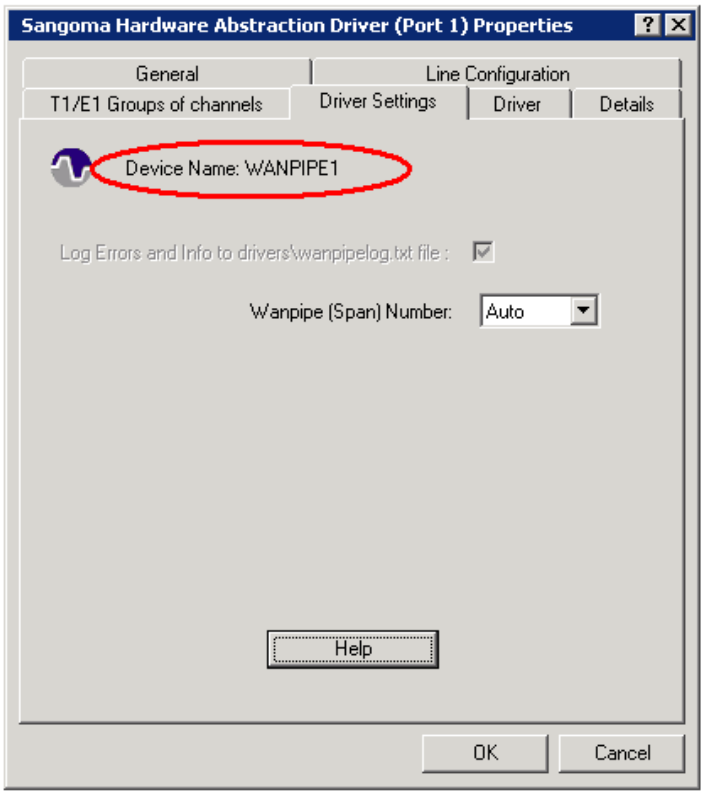
   c) If necessary, click the plus sign (+) beside "Network adapters" to expand the category. You should see a list of the Sangoma AFT device drivers installed.

   In the figure below, four drivers or interfaces are listed. To obtain the wanpipe number of each driver, use the Properties dialog.



2. Right-click on an interface and select **Properties** from the action menu.
3. In the Properties dialog, select the **Driver Settings** tab.
4. Determine the wanpipe number by looking at the Device Name at the top left corner of the display. The wanpipe number is the number that ends the device name.

CAUTION: A given wanpipe number can be used by one PSTN interface *only*.



In the example provided, the wanpipe number is “WANPIPE1”.In the PSTN configuration file, set the wanpipe number using the “*wanpipe*” attribute.

For example:

```
<sangoma id="XYZspan1"
    wanpipe="1"
    mediaType="T1"
    framing="ESF"
    lineEncoding="B8ZS"
    LBO="0dB"
    clocking="TERMINAL"
    loopbackMode="DISABLED">
   <BChannels defaultPcmLaw="PCMU"
          voicePacketLengthInMs="20">
   <VoiceQualityEnhancement>
    <EchoCancellation mode="NORMAL"
                comfortNoiseMode="NORMAL"
                tailDisplacementInMs="0"
                doubleTalkBehavior="OPTIMAL"/>
    <AcousticEchoCancellation mode="NORMAL"/>
    <LevelControl direction="ALL"
              mode="AUTOMATIC"
```

```
                targetLevelIndBM0="-20"/>
    <BackgroundNoiseAttenuation direction="FROM_PSTN"
                        mode="ENABLED"
                        attenuationIndB="-18" />
    <DTMFRemoval direction="FROM_PSTN"
              mode="DISABLED"/>
   </VoiceQualityEnhancement>
  </BChannels>
  <DChannel channelIndex="24"
          mtuSize="2048"/>
</sangoma>
```

Step 3: Set the media type to T1 or E1

The transmission media is set using the "*mediaType*" attribute. Permissible values include:

- **T1:** A 1.44-Mbps digital transmission link used in North America and Japan, composed of 23 B channels (voice/data channel) and one D channel (call control channel).
- **E1:** A 2.048-Mbps digital transmission link used in Europe, Australia and most of the rest of the world, composed of 30 B channels and one D channel.

NOTE: In ISDN, the channel on which the voice, data, and fax is carried is called a bearer (or *B channel*). The channel on which the signal is carried is called a data or a control channel (*D channel*).

To set the "mediaType" attribute:

1. For each interface, determine if the interface is a T1 or an E1.
2. Configure the interface using the "*mediaType*" attribute.

   In the first example below, the "*mediaType*" is set to T1.

```
<sangoma id="XYZspan1"
    wanpipe="1"
    mediaType="T1"
    framing="ESF"
    lineEncoding="B8ZS"
    LBO="0dB"
    clocking="TERMINAL"
    loopbackMode="DISABLED">
   <BChannels defaultPcmLaw="PCMU"
         voicePacketLengthInMs="20">
   <VoiceQualityEnhancement>
    <EchoCancellation mode="NORMAL"
                comfortNoiseMode="NORMAL"
                tailDisplacementInMs="0"
                doubleTalkBehavior="OPTIMAL"/>
    <AcousticEchoCancellation mode="NORMAL"/>
    <LevelControl direction="ALL"
              mode="AUTOMATIC"
              targetLevelIndBM0="-20"/>
    <BackgroundNoiseAttenuation direction="FROM_PSTN"
                        mode="ENABLED"
```

```
                    attenuationIndB="-18" />
    <DTMFRemoval direction="FROM_PSTN"
            mode="DISABLED"/>
   </VoiceQualityEnhancement>
  </BChannels>
  <DChannel channelIndex="24"
        mtuSize="2048"/>
</sangoma>
```

In this second example, the "*mediaType*" is set to E1.

```
<sangoma id="XYZspan1"
    wanpipe="1"
    mediaType="E1"
    framing="ESF"
    lineEncoding="B8ZS"
    clocking="TERMINAL"
    E1Signaling="CSS"
    E1Termination="120ohm"
    loopbackMode="DISABLED">
   <BChannels defaultPcmLaw="PCMA"
          voicePacketLengthInMs="20">
   <VoiceQualityEnhancement>
    <EchoCancellation mode="NORMAL"
                comfortNoiseMode="NORMAL"
                tailDisplacementInMs="0"
                doubleTalkBehavior="OPTIMAL"/>
    <AcousticEchoCancellation mode="NORMAL"/>
    <LevelControl direction="ALL"
              mode="AUTOMATIC"
              targetLevelIndBM0="-20"/>
    <BackgroundNoiseAttenuation direction="FROM_PSTN"
                        mode="ENABLED"
                        attenuationIndB="-18" />
    <DTMFRemoval direction="FROM_PSTN"
             mode="DISABLED"/>
   </VoiceQualityEnhancement>
  </BChannels>
  <DChannel channelIndex="16"
        mtuSize="2048"/>
</sangoma>
```

CAUTION: The "*mediaType*" attribute of the PSTN interfaces residing on the same Sangoma board must specify the same media type. In other words, you cannot mix T1 and E1 on the same Sangoma board. For example, if an application requires two T1 interfaces and two E1 interfaces, then you will need two Sangoma boards with dual interfaces; a single quad interface board will not suffice.

Step 4: Set the framing and line encoding

Framing and line encoding parameters are provided by the carrier or the PBX administrator. It is important to configure these properties properly; otherwise, the interface will never come up and you will receive a red alarm (see Interpreting PSTN alarms on page 88).

The table below provides the valid framing and line encoding values for the different media types (T1 and E1).

Framing and Line Encoding Values		
mediaType	***T1***	***E1***
framing	ESF D4 ESF_Japan	CRC4 NonCRC4
lineEncoding	B8ZS AMI	HDB3 AMI

To set the framing and line encoding:

1. For each interface, assign a valid value to the "*framing*" attribute.
2. For each interface, assign a valid value to the "*lineEncoding*" attribute.

   For example:

```
<sangoma id="XYZspan1"
    wanpipe="1"
    mediaType="T1"
    framing="ESF"
    lineEncoding="B8ZS"
    LBO="0dB"
    clocking="TERMINAL"
    loopbackMode="DISABLED">
   <BChannels defaultPcmLaw="PCMU"
          voicePacketLengthInMs="20">
   <VoiceQualityEnhancement>
    <EchoCancellation mode="NORMAL"
                comfortNoiseMode="NORMAL"
                tailDisplacementInMs="0"
                doubleTalkBehavior="OPTIMAL"/>
    <AcousticEchoCancellation mode="NORMAL"/>
    <LevelControl direction="ALL"
              mode="AUTOMATIC"
```

```
                targetLevelIndBM0="-20"/>
    <BackgroundNoiseAttenuation direction="FROM_PSTN"
                        mode="ENABLED"
                        attenuationIndB="-18" />
    <DTMFRemoval direction="FROM_PSTN"
              mode="DISABLED"/>
   </VoiceQualityEnhancement>
  </BChannels>
  <DChannel channelIndex="24"
          mtuSize="2048"/>
 </sangoma>
```

Step 5: Set the clock source

The clock source of the interface is very important in TDM networks. Usually the clock source is provided by the carrier or the PBX in order to use the same clock sources for every node in the network.

To set the clock source:

1. Determine the clock source. Use the following guidelines:
   a) To use the clock source of the carrier or provider, set the "*clocking*" attribute to TERMINAL.
   b) To use the Gateway to provide the clock source to the other node in the network, set the "*clocking*" attribute to NETWORK.
   c) Always follow the rule wherein one end of a T1 link shall be a clock provider (clocking="NETWORK") and the other shall be a clock consumer (clocking="TERMINAL").
2. For each interface, assign a valid value to the "*clocking*" attribute.

   In the example below, a T1 loopback cable is configured to loopback the traffic from the "XYZ*span1*" interface to the "XYZ*span2*" interface. Note that one interface is configured with the clocking attribute set to "*NETWORK*", while the other is configured to "*TERMINAL*".

```
<sangoma id="XYZspan1"
    wanpipe="1"
    mediaType="T1"
    framing="ESF"
    lineEncoding="B8ZS"
    LBO="0dB"
    clocking="NETWORK"
    loopbackMode="DISABLED">
   <BChannels defaultPcmLaw="PCMU"
          voicePacketLengthInMs="20">
```

```
    <VoiceQualityEnhancement>
     <EchoCancellation mode="NORMAL"
                       comfortNoiseMode="NORMAL"
                       tailDisplacementInMs="0"
                       doubleTalkBehavior="OPTIMAL"/>
     <AcousticEchoCancellation mode="NORMAL"/>
     <LevelControl direction="ALL"
                   mode="AUTOMATIC"
                   targetLevelIndBM0="-20"/>
     <BackgroundNoiseAttenuation direction="FROM_PSTN"
                                 mode="ENABLED"
                                 attenuationIndB="-18" />
     <DTMFRemoval direction="FROM_PSTN"
                  mode="DISABLED"/>
    </VoiceQualityEnhancement>
   </BChannels>
   <DChannel channelIndex="24"
             mtuSize="2048"/>
 </sangoma>

<sangoma id="XYZspan2"
     wanpipe="2"
     mediaType="T1"
     framing="ESF"
     lineEncoding="B8ZS"
     LBO="0dB"
     clocking="TERMINAL"
     loopbackMode="DISABLED">
    <BChannels defaultPcmLaw="PCMU"
               voicePacketLengthInMs="20">
    <VoiceQualityEnhancement>
     <EchoCancellation mode="NORMAL"
                       comfortNoiseMode="NORMAL"
                       tailDisplacementInMs="0"
                       doubleTalkBehavior="OPTIMAL"/>
     <AcousticEchoCancellation mode="NORMAL"/>
     <LevelControl direction="ALL"
                   mode="AUTOMATIC"
                   targetLevelIndBM0="-20"/>
     <BackgroundNoiseAttenuation direction="FROM_PSTN"
                                 mode="ENABLED"
                                 attenuationIndB="-18" />
     <DTMFRemoval direction="FROM_PSTN"
                  mode="DISABLED"/>
    </VoiceQualityEnhancement>
   </BChannels>
```

```
  <DChannel channelIndex="24"
        mtuSize="2048"/>
</sangoma>
```

Step 6: Set T1 cable length

T1 can amplify the output signal to accommodate different cable lengths. This amplification is configured using the "*LBO*" attribute, which stands for Line Built-in Output. The following values are supported:

a) 0dB
b) 7.5dB
c) 15dB
d) 22.5dB
e) 0-110feet
f) 110-220feet
g) 220-330feet
h) 330-440feet
i) 440-550feet
j) 550-660feet

To set the T1 cable length:

1. Configure the T1 cable length by setting the "*LBO*" attribute to one of the permissible values.

   For example:

```
<sangoma id="XYZspan1"
    wanpipe="1"
    mediaType="T1"
    framing="ESF"
    lineEncoding="B8ZS"
    LBO="0dB"
    clocking="NETWORK"
    loopbackMode="DISABLED">
   <BChannels defaultPcmLaw="PCMU"
          voicePacketLengthInMs="20">
   <VoiceQualityEnhancement>
    <EchoCancellation mode="NORMAL"
                comfortNoiseMode="NORMAL"
                tailDisplacementInMs="0"
                doubleTalkBehavior="OPTIMAL"/>
    <AcousticEchoCancellation mode="NORMAL"/>
    <LevelControl direction="ALL"
              mode="AUTOMATIC"
              targetLevelIndBM0="-20"/>
    <BackgroundNoiseAttenuation direction="FROM_PSTN"
                        mode="ENABLED"
                        attenuationIndB="-18" />
    <DTMFRemoval direction="FROM_PSTN"
             mode="DISABLED"/>
   </VoiceQualityEnhancement>
  </BChannels>
  <DChannel channelIndex="24"
        mtuSize="2048"/>
 </sangoma>
```

NOTE: This parameter is ignored by the E1 interface.

Enabling/disabling echo cancellation

Echo is the phenomenon of hearing one's own voice, with a delay. The larger this delay, the more discomfort experienced by the person hearing it. Beyond a certain threshold, conversation is impeded and becomes unnatural and ultimately detrimental to communication.

Echo cancellation devices provide a means to remove echo. The *Internal Telecommunication Union (ITU)* defines echo cancellation as follows:

"*Echo cancellers are voice operated devices placed in the 4-wire portion of a circuit (which may be an individual circuit path or a path carrying a multiplexed signal) and are used for reducing the echo by subtracting an estimated echo from the circuit echo.*" (Telecommunication Standardization Sector Recommendation G.168 [ITU-T G.168])

A key component of echo cancellation is the ability to handle long tail lengths, also referred to as "echo tail", which is defined by G.168 as "*the maximum echo path delay for which an echo canceller is designed to operate*."The Gateway provides an extended 128-ms echo tail buffer, providing robust echo cancellation performance. Note that the echo tail buffer does not impact performance of the Gateway in any way.

The "*BChannels*" sub element of the PSTN configuration file contains attributes that both control the media format and activate the echo cancellation feature on the PSTN interface.

To configure the "*BChannels*" element:

1. Set the the media format of the data coming in or out of the interface using the "*defaultPcmLaw*" attribute. The encoding value is based on country and regional boundaries, as follows:
   - **ulaw (PCMU):** used in North America
   - **A-law (PCMA):** used in Europe and the rest of the world.

   In the example below, the "*defaultPcmLaw*" attribute value is "PCMU" or PCM ulaw encoding. To set the encoding to PCM A-law, you would change the value to "PCMA".

```
<sangoma id="XYZspan1"
    wanpipe="1"
    mediaType="T1"
    framing="ESF"
    lineEncoding="B8ZS"
```

```
    LBO="0dB"
    clocking="NETWORK"
    loopbackMode="DISABLED">
  <BChannels defaultPcmLaw="PCMU"
          voicePacketLengthInMs="20">
   <VoiceQualityEnhancement>
    <EchoCancellation mode="NORMAL"
                comfortNoiseMode="NORMAL"
                tailDisplacementInMs="0"
                doubleTalkBehavior="OPTIMAL"/>
    <AcousticEchoCancellation mode="NORMAL"/>
    <LevelControl direction="ALL"
                mode="AUTOMATIC"
                targetLevelIndBM0="-20"/>
    <BackgroundNoiseAttenuation direction="FROM_PSTN"
                        mode="ENABLED"
                        attenuationIndB="-18" />
    <DTMFRemoval direction="FROM_PSTN"
                mode="DISABLED"/>
   </VoiceQualityEnhancement>
  </BChannels>
  <DChannel channelIndex="24"
          mtuSize="2048"/>
 </sangoma>
```

2. Indicate the size of the RTP packets coming in and out of the Gateway by using the "*voicePacketLengthInMs*" attribute.

   In the example below, the application is expecting to use 20 ms (160 byte) RTP (Real-time Transport Protocol) streams; therefore, the "*voicePacketLengthInMs*" attribute value is "20".

```
<sangoma id="XYZspan1"
    wanpipe="1"
    mediaType="T1"
    framing="ESF"
    lineEncoding="B8ZS"
    LBO="0dB"
    clocking="NETWORK"
    loopbackMode="DISABLED">
  <BChannels defaultPcmLaw="PCMU"
          voicePacketLengthInMs="20">
   <VoiceQualityEnhancement>
    <EchoCancellation mode="NORMAL"
                comfortNoiseMode="NORMAL"
                tailDisplacementInMs="0"
                doubleTalkBehavior="OPTIMAL"/>
```

```
    <AcousticEchoCancellation mode="NORMAL"/>
    <LevelControl direction="ALL"
                  mode="AUTOMATIC"
                  targetLevelIndBM0="-20"/>
    <BackgroundNoiseAttenuation direction="FROM_PSTN"
                        mode="ENABLED"
                        attenuationIndB="-18" />
    <DTMFRemoval direction="FROM_PSTN"
                 mode="DISABLED"/>
   </VoiceQualityEnhancement>
  </BChannels>
  <DChannel channelIndex="24"
          mtuSize="2048"/>
</sangoma>
```

CAUTION: This parameter directly affects the CPU usage of the Gateway. The larger this value, the lower the CPU usage. For detailed information on CPU usage, see Provisioning on page 18.

3. Enable or disable echo cancellation as required, using the "*mode*" attribute of the "*EchoCancellation*" element. Use the following values:
   a) **NORMAL:** To enable echo cancellation on all voice channels (B channels). In this mode, the Gateway will clean the echo of the voice stream sent toward the RTP side of the Gateway.
   b) **SPEECH_RECOGNIZER_FRIENDLY:** To enable echo cancellation on all voice channels (B channels) for application where speech recognizer engines are used. In this mode, the Gateway will clean the echo of the voice stream sent toward the RTP side of the Gateway.
   c) **DISABLED:** To disable echo cancellation on all voice channels (B channels). See the example below.

NOTE: Echo cancellation is disabled automatically upon detection of a fax or modem.

```
<sangoma id="XYZspan1"
    wanpipe="1"
    mediaType="T1"
    framing="ESF"
    lineEncoding="B8ZS"
    LBO="0dB"
    clocking="NETWORK"
```

```
    loopbackMode="DISABLED">
  <BChannels defaultPcmLaw="PCMU"
          voicePacketLengthInMs="20">
   <VoiceQualityEnhancement>
    <EchoCancellation mode="NORMAL"
                comfortNoiseMode="NORMAL"
                tailDisplacementInMs="0"
                doubleTalkBehavior="OPTIMAL"/>
    <AcousticEchoCancellation mode="NORMAL"/>
    <LevelControl direction="ALL"
              mode="AUTOMATIC"
              targetLevelIndBM0="-20"/>
    <BackgroundNoiseAttenuation direction="FROM_PSTN"
                        mode="ENABLED"
                        attenuationIndB="-18" />
    <DTMFRemoval direction="FROM_PSTN"
             mode="DISABLED"/>
   </VoiceQualityEnhancement>
  </BChannels>
  <DChannel channelIndex="24"
        mtuSize="2048"/>
</sangoma>
```

CAUTION: Currently, the echo cancellation device is used to perform *DTMF (Dual-Tone Multi-Frequency)* detection; thus, this parameter must be set to “NORMAL” or “SPEECH_RECOGNIZER_FRIENDLY” in order to enable RFC 2833 events transmission.

Configuring ISDN signaling

ISDN (Integrated Services Digital Network) is a set of CCITT/ITU standards for digital transmission over ordinary telephone copper wire as well as over other media. There are two levels of service: *Basic Rate Interface (BRI)*, intended for the home and small enterprise, and Primary Rate Interface (PRI), for larger users. Both rates include a number of B channels and a D channel. Each B channel carries data, voice, and other services, whereas the D channel carries control and signaling information.

In general, ISDN is the integration of both analog or voice data together with digital data over the same network.

Signaling

ISDN uses signaling protocols to set up, control and terminate calls. This signaling protocol is formally specified in ITU-T Q.921 and Q.931.

ISDN uses two methods of signaling, in-band and out-of-band. The two methods indicate whether various signals travel on the same channel (or *band*) with voice calls or data (in-band), or whether those signals travel on a separate channel (out-of-band). Examples are provided below:

- **In-Band:** T1 Channel-associated signaling (CAS), which is based on Robbed-bit signaling, uses bits from specified frames in the user data channel for signaling.
- **Out-of-Band:** ISDN-Primary Rate Interface (PRI) uses the D channel for signaling and the B channels for user data. E1 Channel Associated Signaling (CAS) uses E1 time slot 16 (the D channel).

International variants of ISDN

The organization primarily responsible for producing the ISDN standards is the ITU-T (formerly the CCITT or International Telephone and Telegraph Consultative Committee). Prior to the publication of the first set of ISDN recommendations in 1984 (Red Books), various geographical areas had developed different versions of ISDN. This resulted in the ITU-T recommendation of a common ISDN standard for all countries, in addition to allocated variants defined for each country.

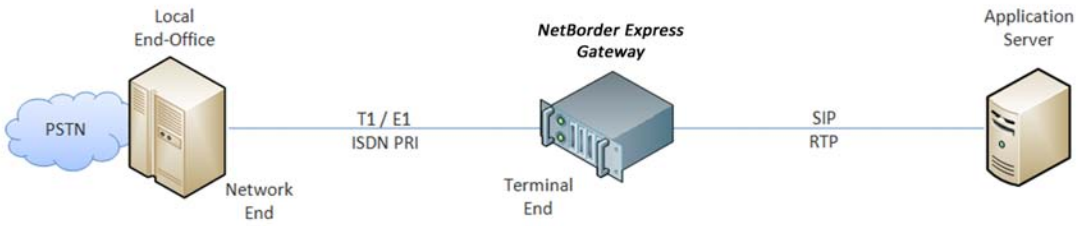
The Gateway supports the following ISDN variants:

- **5ESS (AT&T):** This variant is used in the USA by AT&T.
- **Euro ISDN (ETSI):** This variant is to be adopted by all of the European countries.
- **National ISDN 2 (Bellcore):** This variant is used in the USA by Bellcore.
- **DMS100:** This variant represents Nortel's implementation of National ISDN-1.

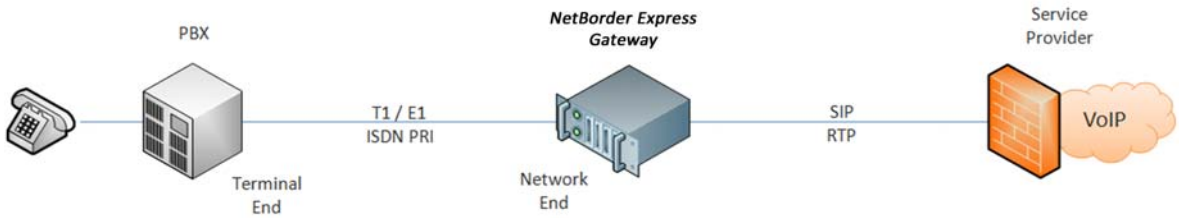
Please consult us if your require additional variants.

Network termination

A media gateway often connects SIP terminals to the PSTN. In such cases, the media gateway must be seen in the PSTN network as an ISDN terminal equipment (TE) or terminal end device.



For example, in some circumstances, the Gateway might be used to connect a TDM PBX to a VoIP carrier. In this scenario, the media gateway must be seen by the TDM PBX; therefore, the media gateway must be configured as an ISDN network termination (NT) or network end device.



Non-Facility Associated Signalling

In the North American telecommunications system, a T1 typically carries 24 individual timeslots. Each timeslot in turn carries a single telephone call. When a T1 circuit is used to carry Primary Rate ISDN one of the timeslots is used to carry the D channel.

In an NFAS (Non-Facility Associated Signalling) configuration, multiple T1 share a single D channel. There is an upper limit of 20 T1 in a single NFAS group.

A single Primary Rate ISDN circuit is sometimes described as 23B + D. There are 23 bearer channels carrying voice, and one D channel carrying the Common Channel Signalling.

A single full NFAS group can then be described as 479B + D (20 T1).

Creating an ISDN group

In the PSTN configuration file, the ISDN "*groups*" element contains the ISDN signaling parameters (for example, the ISDN variant and termination side) for the PSTN interfaces.

Keep in mind the following guidelines when creating ISDN groups:

- An ISDN group can contain only one interface, and an interface can appear in only one ISDN group.
- For each PSTN interface, the configuration file must contain one ISDN group.

Let's say that we have two T1 interfaces, which we want to connect to our carrier. The carrier has told us to configure our interfaces in T1, extended super frame (ESF) framing, with B8ZS line encoding, and with ISDN protocol for the Nortel DMS-100 variant.

In this case, our configuration will contain two sangoma interfaces called "Bell_*span1*" and "Bell_*span2*" configured with *mediaType*="T1", *framing*="ESF", and *lineEncoding*="B8ZS". Since we are connecting to a carrier, the clocking parameters will be set to "TERMINAL".

See the example below:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- RelaxNG shema=pstn-config.rng -->
<pstnConfig version="1.0">

 <!-- ... -->

 <!-- All PSTN interfaces -->
 <interfaces>
  <sangoma id="Bell_span1"
      wanpipe="1"
      mediaType="T1"
      framing="ESF"
      lineEncoding="B8ZS"
      LBO="0dB"
      clocking="TERMINAL"
      loopbackMode="DISABLED">
     <BChannels defaultPcmLaw="PCMU"
            voicePacketLengthInMs="20">
     <VoiceQualityEnhancement>
      <EchoCancellation mode="NORMAL"
                  comfortNoiseMode="NORMAL"
                  tailDisplacementInMs="0"
```

```
                    doubleTalkBehavior="OPTIMAL"/>
      <AcousticEchoCancellation mode="NORMAL"/>
      <LevelControl direction="ALL"
                  mode="AUTOMATIC"
                  targetLevelIndBM0="-20"/>
      <BackgroundNoiseAttenuation direction="FROM_PSTN"
                            mode="ENABLED"
                            attenuationIndB="-18" />
      <DTMFRemoval direction="FROM_PSTN"
                 mode="DISABLED"/>
    </VoiceQualityEnhancement>
  </BChannels>
  <DChannel channelIndex="24"
          mtuSize="2048"/>
 </sangoma>
 <sangoma id="Bell_span2"
     wanpipe="2"
     mediaType="T1"
     framing="ESF"
     lineEncoding="B8ZS"
     LBO="0dB"
     clocking="TERMINAL"
     loopbackMode="DISABLED">
    <BChannels defaultPcmLaw="PCMU"
           voicePacketLengthInMs="20">
    <VoiceQualityEnhancement>
      <EchoCancellation mode="NORMAL"
                  comfortNoiseMode="NORMAL"
                  tailDisplacementInMs="0"
                  doubleTalkBehavior="OPTIMAL"/>
      <AcousticEchoCancellation mode="NORMAL"/>
      <LevelControl direction="ALL"
                  mode="AUTOMATIC"
                  targetLevelIndBM0="-20"/>
      <BackgroundNoiseAttenuation direction="FROM_PSTN"
                            mode="ENABLED"
                            attenuationIndB="-18" />
      <DTMFRemoval direction="FROM_PSTN"
                 mode="DISABLED"/>
    </VoiceQualityEnhancement>
  </BChannels>
  <DChannel channelIndex="24"
          mtuSize="2048"/>
 </sangoma>
</interfaces>
```

```
<!-- ... -->
```

<!-- ... --> above indicates that elements have either been omitted, for clarity, or are yet to be configured.

To create ISDN groups:

1. Create the required number of interface ISDN groups in your PSTN configuration file, one for each interface. Use the "*group*" element, which is a child of *<callControl><isdn><groups>*.
2. Assign a unique identifier to each group using the "*ID*" attribute.

   Using our example above, you would create two groups, one for each of the sangoma interfaces, as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<pstnConfig version="1.0">

 <!-- ... -->

 <!-- Call control parameters -->
 <callControl>

  <!-- Resources groups . . . -->

  <!-- ISDN parameters -->
  <isdn>

   <!— Single interfaces ISDN groups -->
   <groups>

    <group ID="Bell1_ISDN" . . . >
     <!-- ... -->
    </group>

    <group ID="Bell2_ISDN" . . . >
     <!-- ... -->
    </group>

   </groups>
  </isdn>
 </callControl>
</pstnConfig>
```

3. Assign a switch variant to the group using the "*switchVariant*" attribute.

   In our example, we need to set the switch variant to "*DMS100*", as we

are using the Nortel DMS-100 variant.

```
<group ID="Bell2_ISDN"
       termination="TERMINAL"
       switchVariant="DMS100"
       BChannelNegotiation="EXCLUSIVE">
 <member interfaceID="Bell_span2">
  <BChannels>
   <BChannelRange firstChannel="1" lastChannel="23">
    <resourcesGroup>Bell1 bidir</resourcesGroup>
   </BChannelRange>
  </BChannels>
 </member>
</group>
```

4. Assign the termination side to "*TERMINAL*", since we are connecting the Gateway to a PSTN switch. Set the "*termination*" attribute as follows:

```
<group ID="Bell2_ISDN"
       termination="TERMINAL"
       switchVariant="DMS100"
       BChannelNegotiation="EXCLUSIVE">
 <member interfaceID="Bell_span2">
  <BChannels>
   <BChannelRange firstChannel="1" lastChannel="23">
    <resourcesGroup>Bell1 bidir</resourcesGroup>
   </BChannelRange>
  </BChannels>
 </member>
</group>
```

5. Ensure that the "*BChannelNegotiation*" attribute is set to "*EXCLUSIVE*".
6. Use the "*member*" element to indicate the PSTN interface member of the group. This value must match the ID of the interface listed in the "*interfaces*" element (*<PstnConfig><interfaces>*).

   In our example, the "*Bell2_ISDN*" group will run on the "*Bell_span2*" PSTN interface, as follows:

```
<group ID="Bell2_ISDN"
       termination="TERMINAL"
       switchVariant="DMS100"
       BChannelNegotiation="EXCLUSIVE">
 <member interfaceID="Bell_span2" >
  <BChannels>
```

```
        <BChannelRange firstChannel="1" lastChannel="23">
          <resourcesGroup>Bell1 bidir</resourcesGroup>
        </BChannelRange>
      </BChannels>
    </member>
  </group>
```

As you can see from our example, the active B channels of the ISDN group are specified using the “BChannels” element. This element consists of a list of “*BChannelRange*” sub elements.

7. Define a range of channels using the “*BChannelRange*” element. Use the “*firstChannel*” attribute to specify the first channel of the range, and the “*lastChannel*” element to specify the last channel.

   In our example, we have created a single “*BChannelRange*” element with the first channel set to “1” and the last channel set to “23”, as follows:

```
    <group ID="Bell2_ISDN"
           termination="TERMINAL"
           switchVariant="DMS100"
           BChannelNegotiation="EXCLUSIVE">
     <member interfaceID="Bell_span2">
      <BChannels>
       <BChannelRange firstChannel="1" lastChannel="23">
        <resourcesGroup>Bell1 bidir</resourcesGroup>
       </BChannelRange>
      </BChannels>
     </member>
    </group>
```

8. Use the "*resourcesGroup*" element to specify the resources group to which this channel belongs.

   For example, if we have a "*resourcesGroup*" element with an ID of "*Bell1 bidir*" to which we want to associate our B channel, we would specify the the ID of the resources group. For example:

```
<!--Resources groups -->
  <resourcesGroups>

    <!-- ... -->
   <resourcesGroup ID="Bell1 bidir"
                direction="BIDIR"
                outboundHuntingScheme="REVERSE_LINEAR"
                inboundHuntingScheme="LINEAR"/>

  </resourcesGroups> <!-- End of Resources groups -->

  <!-- ISDN parameters -->
  <isdn>


   <!— Single interfaces ISDN groups -->
   <groups>

    <!-- ... -->
    <group ID="Bell2_ISDN"
           termination="TERMINAL"
           switchVariant="DMS100"
           BChannelNegotiation="EXCLUSIVE">
     <member interfaceID="Bell_span2">
      <BChannels>
       <BChannelRange firstChannel="1" lastChannel="23">
        <resourcesGroup>Bell1 bidir</resourcesGroup>
```

```
</BChannelRange>
        </BChannels>
      </member>
    </group>
```

If your application needs to assign the lower B channels (for instance, 1-12) resources for inbound calls, and the upper B channels (for instance, 13-23) resources for output calls, the "*BChannels*" element should contain two "*BChannelsRange*" elements.

See the example below.

```
    <group ID="Bell2_ISDN"
          termination="TERMINAL"
          switchVariant="DMS100"
          BChannelNegotiation="EXCLUSIVE">
      <member interfaceID="Bell_span2">
        <BChannels>
          <BChannelRange firstChannel="1" lastChannel="12">
            <resourcesGroup>Bell1 inbound</resourcesGroup>
          </BChannelRange>
          <BChannelRange firstChannel="13" lastChannel="23">
            <resourcesGroup>Bell1 outbound</resourcesGroup>
          </BChannelRange>
        </BChannels>
      </member>
    </group>
```

For more information on Resources Groups, see Creating and configuring a resources group on page 135.

Creating an ISDN NFAS group

In the PSTN configuration file, the ISDN "nfasG*roups*" element contains the ISDN signaling parameters for all NFAS groups(for example, the ISDN variant and termination side) for the PSTN interfaces.

Keep in mind the following guidelines when creating ISDN groups:

- An ISDN NFAS group can contain one or more interface (up to 20).
- An interface can appear in only one ISDN group.
- An ISDN NFAS group can contains only one primary interface. The primary interface specifies where the D-Channel resides.
- For each PSTN primary interface, the configuration file must specifies the D-Channel.

Let's say that we have two T1 interfaces, which we want to connect to our carrier. The carrier has told us to configure our interfaces in T1, extended super frame (ESF) framing, with B8ZS line encoding, and with ISDN protocol for the Lucent 5ESS variant in NFAS mode with a single D-Channel for the two interfaces.

In this case, our configuration will contain two sangoma interfaces called "Bell_*span1*" and "Bell_*span2*" configured with *mediaType*="T1", *framing*="ESF", and *lineEncoding*="B8ZS". Since we are connecting to a carrier, the clocking parameters will be set to "TERMINAL". The first interface defines the D-Channel on channel 24, while the second interface don't define a D-Channel.

See the example below:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- RelaxNG shema=pstn-config.rng -->
<pstnConfig version="1.0">

 <!-- ... -->

 <!-- All PSTN interfaces -->
 <interfaces>
  <sangoma id="Bell_span1"
      wanpipe="1"
      mediaType="T1"
      framing="ESF"
      lineEncoding="B8ZS"
      LBO="0dB"
      clocking="TERMINAL"
      loopbackMode="DISABLED">
     <BChannels defaultPcmLaw="PCMU"
            voicePacketLengthInMs="20">
     <VoiceQualityEnhancement>
      <EchoCancellation mode="NORMAL"
                  comfortNoiseMode="NORMAL"
                  tailDisplacementInMs="0"
                  doubleTalkBehavior="OPTIMAL"/>
      <AcousticEchoCancellation mode="NORMAL"/>
      <LevelControl direction="ALL"
                mode="AUTOMATIC"
                targetLevelIndBM0="-20"/>
      <BackgroundNoiseAttenuation direction="FROM_PSTN"
                        mode="ENABLED"
                        attenuationIndB="-18" />
      <DTMFRemoval direction="FROM_PSTN"
                mode="DISABLED"/>
```

```
      </VoiceQualityEnhancement>
    </BChannels>
    <DChannel channelIndex="24"
            mtuSize="2048"/>
  </sangoma>
  <sangoma id="Bell_span2"
      wanpipe="2"
      mediaType="T1"
      framing="ESF"
      lineEncoding="B8ZS"
      LBO="0dB"
      clocking="TERMINAL"
      loopbackMode="DISABLED">
     <BChannels defaultPcmLaw="PCMU"
             voicePacketLengthInMs="20">
     <VoiceQualityEnhancement>
      <EchoCancellation mode="NORMAL"
                  comfortNoiseMode="NORMAL"
                  tailDisplacementInMs="0"
                  doubleTalkBehavior="OPTIMAL"/>
      <AcousticEchoCancellation mode="NORMAL"/>
      <LevelControl direction="ALL"
                mode="AUTOMATIC"
                targetLevelIndBM0="-20"/>
      <BackgroundNoiseAttenuation direction="FROM_PSTN"
                          mode="ENABLED"
                          attenuationIndB="-18" />
      <DTMFRemoval direction="FROM_PSTN"
                mode="DISABLED"/>
     </VoiceQualityEnhancement>
    </BChannels>
 </interfaces>

<!-- ... -->
```

`<!-- ... -->` above indicates that elements have either been omitted, for clarity, or are yet to be configured.

To create ISDN NFAS groups:

1. Create the required number of interface ISDN groups in your PSTN configuration file, one for each interface. Use the "nfasG*roup*" element, which is a child of *<callControl><isdn><nfasGroups>*.
2. Assign a unique identifier to each group using the "*ID*" attribute.

   Using our example above, you would create two groups, one for each of the sangoma interfaces, as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<pstnConfig version="1.0">

 <!-- ... -->

 <!-- Call control parameters -->
 <callControl>

  <!-- Resources groups -->

  <!-- ISDN parameters -->
  <isdn>

   <!— FAS ISDN groups -->
   <nfasGoups>

    <nfasGoup ID="Bell1_ISDN" . . . >
     <!-- ... -->
    </nfasGroup>

    <nfasGroup ID="Bell2_ISDN" . . . >
     <!-- ... -->
    </nfasGroup>

   </nfasGroups>
  </isdn>
 </callControl>
</pstnConfig>
```

3. Assign a switch variant to the group using the "*switchVariant*" attribute.

   In our example, we need to set the switch variant to "*DMS100*", as we are using the Nortel DMS-100 variant.

```
    <nfasGroup ID="Bell1_ISDN"
          termination="TERMINAL"
          switchVariant="DMS100"
          BChannelNegotiation="EXCLUSIVE">
       <members>
        <member interfaceID="Bell_span1" DChannelMode="PRIMARY">
         <BChannels>
          <BChannelRange firstChannel="1" lastChannel="23">
           <resourcesGroup>Bell1 bidir</resourcesGroup>
          </BChannelRange>
         </BChannels>
        </member>
```

```
    <member interfaceID="Bell_span2" DChannelMode="DISABLED">
     <BChannels>
      <BChannelRange firstChannel="1" lastChannel="24">
       <resourcesGroup>Bell1 bidir</resourcesGroup>
      </BChannelRange>
     </BChannels>
    </member>
   </members>
  </nfasGroup>
```

4. Assign the termination side to "*TERMINAL*", since we are connecting the Gateway to a PSTN switch. Set the "*termination*" attribute as follows:

```
  <nfasGroup ID="Bell1_ISDN"
       termination="TERMINAL"
       switchVariant="DMS100"
       BChannelNegotiation="EXCLUSIVE">
    <members>
     <member interfaceID="Bell_span1" DChannelMode="PRIMARY">
      <BChannels>
       <BChannelRange firstChannel="1" lastChannel="23">
        <resourcesGroup>Bell1 bidir</resourcesGroup>
       </BChannelRange>
      </BChannels>
     </member>
     <member interfaceID="Bell_span2"  DChannelMode="DISABLED">
      <BChannels>
       <BChannelRange firstChannel="1" lastChannel="24">
        <resourcesGroup>Bell1 bidir</resourcesGroup>
       </BChannelRange>
      </BChannels>
     </member>
    </members>
  </nfasGroup>
```

5. Ensure that the "*BChannelNegotiation*" attribute is set to "*EXCLUSIVE*".
6. Use the "*member*" element to indicate the PSTN interface member of the group. This value must match the ID of the interface listed in the "*interfaces*" element (*<PstnConfig><interfaces>*).

   In our example, the "*Bell1_ISDN*" group will run on the "*Bell_span1*" and PSTN interface, as follows:

```
  <nfasGroup ID="Bell1_ISDN"
       termination="TERMINAL"
```

```
        switchVariant="DMS100"
        BChannelNegotiation="EXCLUSIVE">
      <members>
       <member interfaceID="Bell_span1" DChannelMode="PRIMARY">
        <BChannels>
         <BChannelRange firstChannel="1" lastChannel="23">
          <resourcesGroup>Bell1 bidir</resourcesGroup>
         </BChannelRange>
        </BChannels>
       </member>
       <member interfaceID="Bell_span2" DChannelMode="DISABLED">
        <BChannels>
         <BChannelRange firstChannel="1" lastChannel="24">
          <resourcesGroup>Bell1 bidir</resourcesGroup>
         </BChannelRange>
        </BChannels>
       </member>
      </members>
    </nfasGroup>
```

As you can see from our example, the active B channels of the ISDN group are specified using the “BChannels” element. This element consists of a list of “*BChannelRange*” sub elements.

7. Define a range of channels using the “*BChannelRange*” element. Use the “*firstChannel*” attribute to specify the first channel of the range, and the “*lastChannel*” element to specify the last channel.

   In our example, we have created a single “*BChannelRange*” element with the first channel set to “1” and the last channel set to “23” for the first member interface. Note that we have set the channel range of the second interface from “1” to “24”. The first interface carries the D-Channel (*DChannelMode*="*PRIMARY*")while the second interface don't (*DChannelMode*="*DISABLED*"). This explains we have an additional channel for the second interface.

```
    <nfasGroup ID="Bell2_ISDN"
          termination="TERMINAL"
          switchVariant="DMS100"
          BChannelNegotiation="EXCLUSIVE">
      <members>
       <member interfaceID="Bell_span1" DChannelMode="PRIMARY">
        <BChannels>
         <BChannelRange firstChannel="1" lastChannel="23">
          <resourcesGroup>Bell1 bidir</resourcesGroup>
         </BChannelRange>
        </BChannels>
       </member>
       <member interfaceID="Bell_span2" DChannelMode="DISABLED">
        <BChannels>
         <BChannelRange firstChannel="1" lastChannel="24">
          <resourcesGroup>Bell1 bidir</resourcesGroup>
         </BChannelRange>
        </BChannels>
       </member>
      </members>
    </nfasGroup>
```

8. Use the "*resourcesGroup*" element to specify the resources group to which this channel belongs.

   For example, if we have a "*resourcesGroup*" element with an ID of "*Bell1 bidir*" to which we want to associate our B channel, we would specify the the ID of the resources group. For example:

```
<!--Resources groups -->
  <resourcesGroups>

    <!-- ... -->
   <resourcesGroup ID="Bell1 bidir"
              direction="BIDIR"
              outboundHuntingScheme="REVERSE_LINEAR"
              inboundHuntingScheme="LINEAR"/>

  </resourcesGroups> <!-- End of Resources groups -->

  <!-- ISDN parameters -->
  <isdn>


   <!— NFAS interfaces ISDN groups -->
   <nfasGroups>

    <!-- ... -->
```

```
<nfasGroup ID="Bell1_ISDN"
      termination="TERMINAL"
      switchVariant="DMS100"
      BChannelNegotiation="EXCLUSIVE">
 <members>
  <member interfaceID="Bell_span1" DChannelMode="PRIMARY">
   <BChannels>
    <BChannelRange firstChannel="1" lastChannel="23">
     <resourcesGroup>Bell1 bidir</resourcesGroup>
    </BChannelRange>
   </BChannels>
  </member>
  <member interfaceID="Bell_span2" DChannelMode="DISABLED">
   <BChannels>
    <BChannelRange firstChannel="1" lastChannel="24">
     <resourcesGroup>Bell1 bidir</resourcesGroup>
    </BChannelRange>
   </BChannels>
  </member>
 </members>
</nfasGroup>
```

For more information on Resources Groups, see Creating and configuring a resources group on page 135.

Creating and configuring a resources group

A resources group defines the telephony resources to be used by the Gateway based on the defined routing rules, which in turn specify how those resources will be used by the Gateway.

A resources group has a unique identifier among all resources groups, and this identifier is used in the routing rules to specify which resources group is to be used by a given rule. This identifier is specified via the "*ID*" attribute of the "*resourcesGroup*" element. Therefore, to use the resources of a specific group, a given routing rule will specify the "*resourcesGroup ID*".

To create and configure a resource group:

1. Assign a unique identifier to the resources group using the "ID" attribute, as follows:

```
<resourcesGroup ID='Bell0 bidir"
                direction="BIDIR"
                outboundHuntingScheme="LINEAR"
                inboundHuntingScheme="REVERSE_LINEAR"/>
```

2. Use the "*direction*" attribute to specify the direction for which this group of resources is to be used. A telephony resources group could be used for:

   a) **IN:** Inbound calls or calls coming from the PSTN side of the Gateway.

   b) **OUT:** Outbound calls or calls going to the PSTN side of the Gateway.

   c) **BIDIR:** Both inbound and outbound calls.

   In the example below, the direction has been set to "BIDIR" or bi-directional.

```
<resourcesGroup ID="Bell0 bidir"
                direction="BIDIR"
                outboundHuntingScheme="LINEAR"
                inboundHuntingScheme="REVERSE_LINEAR"/>
```

3. Specify how the Gateway will allocate the resources from that group using the "*outboundHuntingScheme*" and "*inboundHuntingScheme*" attributes. The supported hunting schemes are:

   a) **LINEAR:** The Gateway always allocates the first channel available within the resources group. For example, if the resources group contains B channels 1, 2, 3, 5,..., 23, and B channels 1-5 and 20-23 are in use, the Gateway will use the first channel available, 6.

   b) **REVERSE_LINEAR:** The Gateway always allocates the last channel available within the resources group. For example, if the resources group contains B channels 1, 2, 3, 5,..., 23, and B channels 1-5 and 20-23 are in use, the Gateway will use the last channel available, 19.

   In the example below, the "*outboundHuntingScheme*" is set to "*LINEAR*" and the "*inboundHuntingScheme*" is set to "*REVERSE_LINEAR*".

```
<resourcesGroup ID="Bell0 bidir"
                direction="BIDIR"
                outboundHuntingScheme="LINEAR"
                inboundHuntingScheme="REVERSE_LINEAR"/>
```

Chapter 6: SIP and VoIP

This chapter describes how to configure SIP and RTP, and explains how the Gateway maps its operations to conform to the SIP standard.

This chapter contains the following topics:

SIP configuration

This section describes how to configure SIP parameters, as well as how to register the Gateway to a third party SIP registrar.

For information on how the Gateway maps its operations to conform to the SIP standard, see SIP application guidelines on page 145.

Basic parameters

This section is in development.

SIP register configuration

The Gateway can be configured to register itself to a third party SIP registrar. The registration configuration is handled via a separate XML configuration file.

The location and name of this configuration file is provided via the global configuration property *Netborder.sip.clientRegistration.configFile*. For instance, in the example below, the SIP register configuration file is located at *C:\Program Files\Netborder\Express\Gateway\config\sip-client-registration.xml*.

```
# SIP client registration configuration file

Netborder.sip.clientRegistration.configFile=C:\Program Files\Netborder\
Express\Gateway\\sip-client-registration.xml
```

If this property is not set, which is the default behaviour, then no registration will be performed by the Gateway. If this property is set, and a valid file exists, all parameters must be valid and filled (that is, no default values are provided).

The Gateway registers to the registrar on service startup, according to the parameters set in the registration configuration file, and un-registers on service shutdown.

CAUTION: If the Gateway is configured to register with a SIP registrar, and is waiting for a reply from a registrar that is particularly slow or down, it is possible that a service shutdown request times out in Windows before the operation (register or unregister) can be completed.

Registration configuration file

Registration configuration is defined using an XML format. The following table provides a list of the required elements along with a brief description.

SIP Registration Configuration Parameters	
Elements	***Description***
<client_registration_config>	Umbrella element within which all of the registration properties are defined.
<registration_entry>	Umbrella element for every registration entry to be provided to the registrar.
<address_of_record_uri>	A logical SIP URI to be supplied by the Gateway to the registrar. When applications send a request to the registrar for that URI, the Gateway will be contacted using the contact URI provided in the parameter below.
<contact>	Umbrella element for contact information.
<uri>	The URI where the Gateway can be reached for the particular address of record.
<transport>	SIP transport (tcp or udp) to be used when someone wants to use the current contact.
<registrar>	Umbrella element for the registrar parameters.
<uri>	URI of the registrar.
<transport>	SIP transport (tcp or udp) to exchange with the registrar.
<retry_interval_sec>	The interval in seconds during which the Gateway will retry registering (when the registrar does not respond).
<expiration_sec>	Time in seconds after which the registration expires.

Below is a sample registration configuration file:

```
<? xml version="1.0" encoding="UTF-8" ?>
<!--
 Example of a client registration configuration file.
In this example, two SIP register requests, one for Bob and one
for Laura, will be sent by the gateway to the registrar at company.com.
Replace with the relevant information based on your needs.
 -->
<client_registration_config>

 <!-- Registration entry #1 -->
 <registration_entry>
  <address_of_record_uri>sip:Bob@company.com</address_of_record_uri>
  <contact>
   <uri>sip:1014@192.168.0.1:5066</uri>
   <transport>udp</transport>
```

```
  </contact>
</ registration_entry >

<!-- Registration entry #2 -->
<registration_entry>
  <address_of_record_uri>sip:Laura@company.com</address_of_record_uri>
  <contact>
    <uri>sip:1015@192.168.0.1:5066</uri>
    <transport>udp</transport>
  </contact>
</ registration_entry >

<!-- Registrar -->
<registrar>
  <uri>sip:company.com</uri>
  <transport>udp</transport>
  <retry_interval_sec>300</retry_interval_sec>
</registrar>

<expiration_sec>3600</expiration_sec>

</client_registration_config>
```

RTP configuration

The Gateway supports the Real-time Transport Protocol (RTP), as well as the *RTP Control Protocol (RTCP)*, both of which are defined in RFC 3550.

According to the IETF, "*RTP provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services*."

The data transport is augmented by a control protocol (RTCP) to allow monitoring of the data delivery. RTCP partners RTP in the delivery and packaging of multimedia data, but does not transport any data itself. RTP and RTCP are designed to be independent of the underlying transport and network layers.

The Gateway can detect DTMF events on the PSTN side and generate RFC 2833 events toward the VoIP side of the call, or, alternatively, regenerate the DTMF signal on the voice stream on the PSTN side of the call.

Note that the Gateway does **not** support the following:

Voice Activity Detection (VAD): The Gateway continues to stream RTP packets even during periods of silence.

Comfort Noise Generation (CNG): The Gateway generates fixed silence pattern to the PSTN side of the call when no RTP packets are received on the VoIP side of the call.

Configuring the jitter buffer

In VoIP, a jitter buffer is a shared data area where voice packets can be collected, stored and sent to the voice processor in evenly spaced intervals. Variations in packet arrival time, called jitter, can occur because of network congestion, timing drift, or route changes. The jitter buffer, which is located at the receiving end of the voice connection, deliberately delays the arriving packets so that the end user experiences a clear connection with very little sound distortion.

There are two kinds of jitter buffers:

- **static jitter buffer:** hardware-based and configured by the manufacturer
- **dynamic jitter buffer:** software-based and configured by the network administrator to adapt to changes in the network's delay.

The Gateway supports dynamic jitter buffers. The size of all jitter buffers is configurable globally via the following parameters of the global configuration file (*gw.properties*):

- *Netborder.media.rtp.initialDelayMs*
- *Netborder.media.rtp.maxDelayMs*

Netborder.media.rtp.initialDelayMs

The jitter buffer transforms the variable delay introduced by the transmitted packets and/or the network into a fixed delay inside the Gateway. It holds the first packet received for a period of time before it plays it out. This holding period is known as the initial play out delay.

The parameter *Netborder.media.rtp.initialDelayMs* specifies this delay in milliseconds. Note that this parameter does not include the delay added by the Sangoma hardware.

Netborder.media.rtp.maxDelayMs

If packets are held for too short a time, variations in delay can cause the buffer to under-run and introduce gaps in the voice stream. On the other hand, if the sample is held for too long, the buffer can overrun, and the dropped packets again cause gaps in the voice stream. Moreover, if packets are held too long, the overall delay can become unacceptable.

The parameter *Netborder.media.rtp.maxDelayMs* specifies the maximum delay in milliseconds introduced by the jitter buffer. Again, this value does not include the delay added by the Sangoma hardware.

NOTE: If echo cancellation is required, the overall delay should not exceed 128 milliseconds.

To change these parameters, see Editing the global configuration file on page 68.

Enabling/disabling DTMF relay and regeneration

To enable RFC 2833 generation toward IP, and DTMF regeneration from RFC 2833 toward PSTN, make sure the following parameter is set to “True” in the *gw.properties* file:

- *Netborder.media.rtp.rfc2833Supported*

The above assumes that RFC 2833 is enabled for a particular session only if the remote terminal supports it.

To disable the above parameter, be sure to set the value to “False”.

CAUTION: Since DTMF detection is performed by the echo canceller device, RFC2833 generation toward IP, requires the echo canceller device to be enabled. See Enabling/disabling echo cancellation on page 114.

SIP application guidelines

The Gateway is meant to front-end SIP-based applications such as SIP phones, SIP-based media servers, and third-party call control applications. In this section, a number of guidelines and examples are provided to help you get the most out of the Gateway system.

The Gateway complies with RFC 3261. This section highlights the issues and compatibility topics related specifically to telephony, and explains how the Gateway maps its operations to conform to the SIP standard.

This section includes the following topics:

- Call flow examples and issues for both PSTN-initiated calls (page 145) and SIP-initiated calls (page 154).
- SIP status codes including ISDN to SIP mapping and SIP to ISDN mapping, beginning on page 157.

PSTN-initiated calls

In a large number of cases, automated or otherwise, calls come into the Gateway from the traditional telephony network and need to be routed to the appropriate SIP endpoint or application. SIP sessions are established upon reception of incoming PSTN calls, and according to both the call characteristics and the configuration of the Gateway.

Let’s examine the call flow for an incoming PSTN call as it is being routed to the appropriate SIP application.

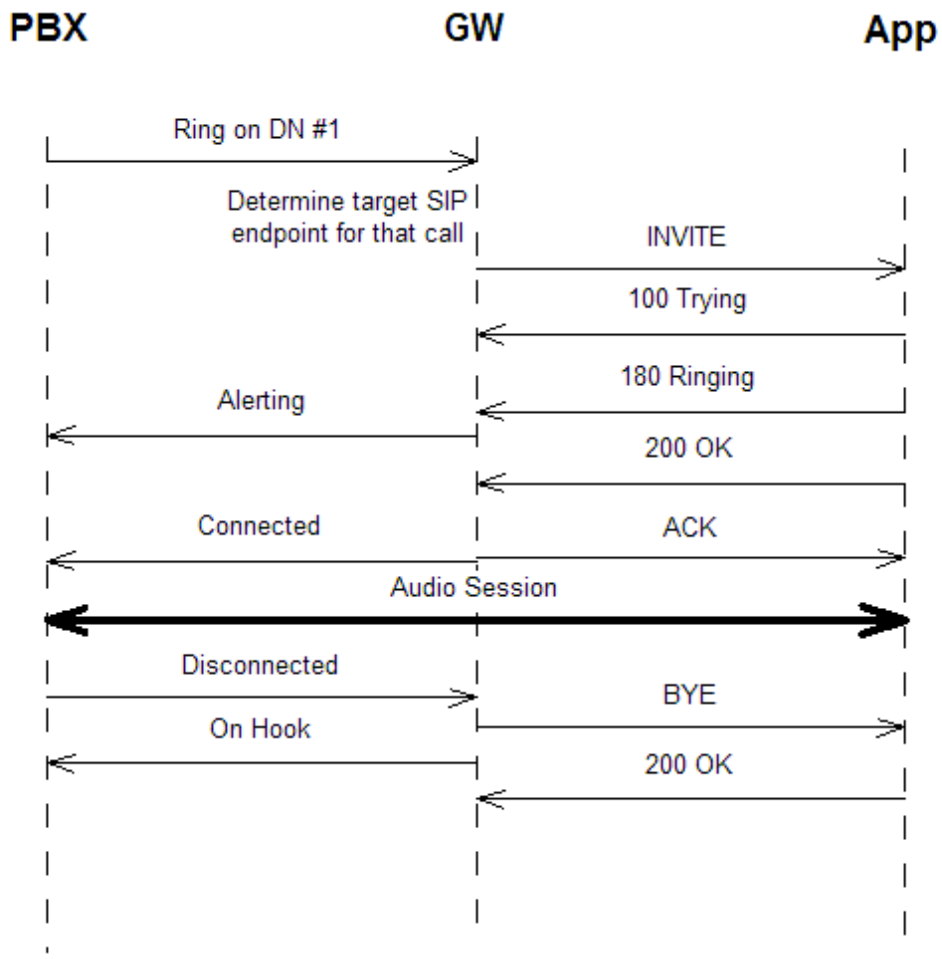
1. Upon notification of an incoming call, the Gateway retrieves all available information related to the call (for example, ANI*/DNIS**) and submits it to the routing rules.
2. Based on the rule that is triggered, an appropriate SIP session is established. Note that the Gateway supports optional provisional responses such as ‘180 Ringing’.
3. Upon reception of a valid provisional response (such as ‘180 Ringing’), the Gateway will send the ‘Alerting’ message on the phone line.
4. The telephony port only goes off-hook once/if the SIP application accepts the call with a message ‘200 OK’.
5. Once the call is established end-to-end from a signaling perspective, the audio sessions are connected together.

6. The session holds until one of the two legs hangs up or a transfer is requested by the SIP application. Once one of the legs hangs up, the other side is immediately disconnected.

   The following diagram illustrates the interactions between the parties.

*ANI (**A**utomatic **N**umber **I**dentification) is a feature that permits the display or capture of the telephone numbers of calling parties. ANI was originally developed for billing purposes. ISDN supports ANI by carrying the calling telephone number in the D channel.
DNIS (D**ialed **N**umber **I**dentification **S**ervice) is a service that determines the number that the caller dialed when accessing the service. This information is useful in determining how to route an inbound call; for example, a service provider might have several toll-free numbers directed to the same call centre and provide unique service based on the number dialed.

PSTN-Initiated Call



If the caller ID/ANI is available in the incoming PSTN call, then using the default routing rules shipped with the Gateway, the caller ID/ANI will automatically be placed in the URI of the 'From' field of the INVITE to the SIP application.

The 'From' URI takes the following format:

- `sip:<caller-ID>@<gateway-host>:<gateway-ua-port>`

  where:

  - `<caller-ID>` is the Caller ID or ANI, as issued by the switch on the PSTN-originated call
  - `<gateway-host>` is the name or IP address of the Gateway host
  - `<gateway-ua-port>` is the port used by the Gateway's SIP User Agent.

The URI is formed such that the application can use it in the 'To' field of an INVITE message to place an outbound call directly to that specific PSTN user. The presentation of the Caller ID/ANI information can be modified by altering the appropriate routing rules. For more information on routing rules, see Chapter 7.

TIP: According to the North-American analog protocol, Caller ID information is sent between the first and second ring. By default, the Gateway is configured to answer calls on the second ring. This is precisely so that the Caller ID information is made available in the 'From' URI received by the SIP application. The format or presentation of the Caller ID information can be modified by changing the corresponding routing rule.

SIP redirection

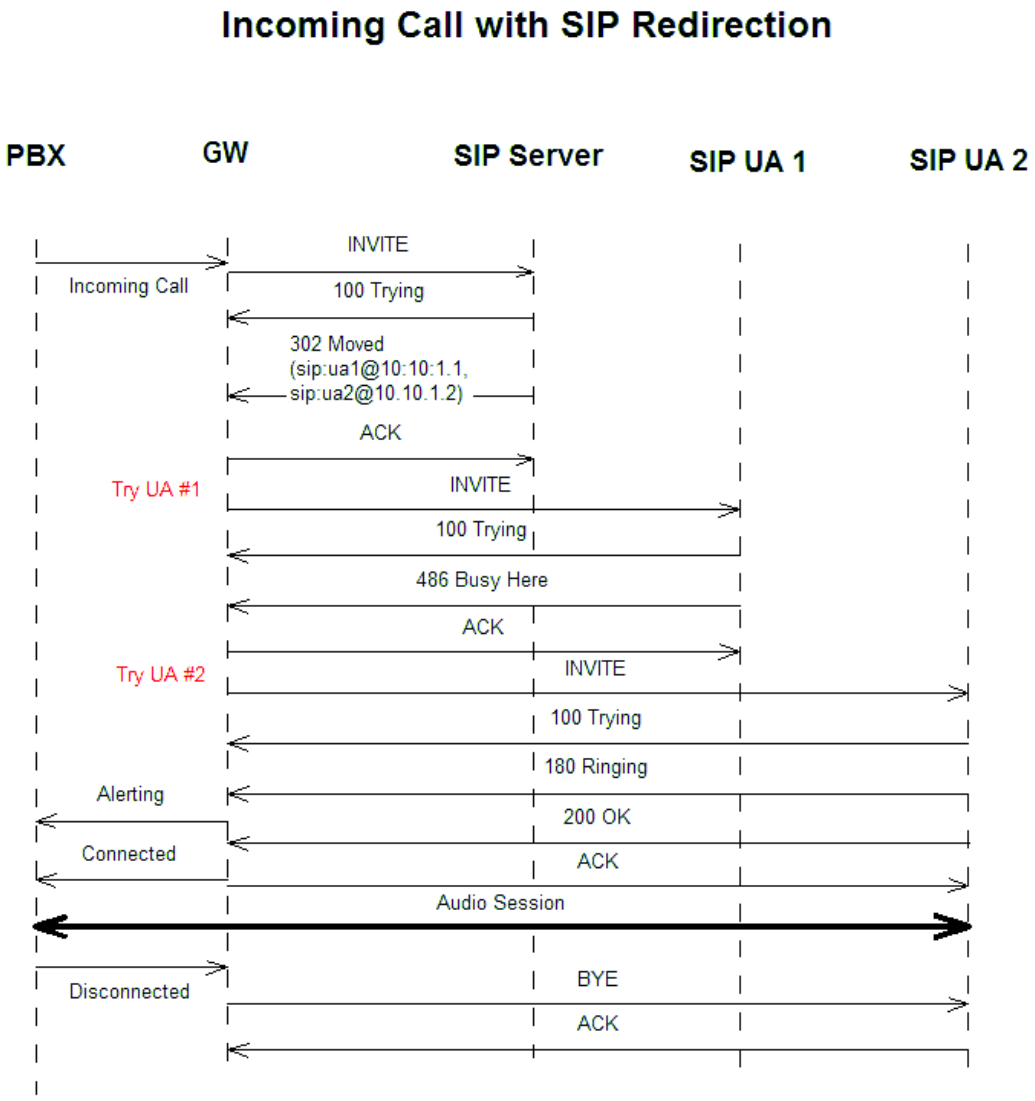
Redirection allows servers to push routing information for a request back in a response to the client. A redirect server is thus a lightweight component that performs basic routing capabilities in the SIP network. It is also the cornerstone of powerful routing applications (such as ACD/CTI in the IP Contact Centre and “follow-me” applications).

The Gateway extends the concept of SIP redirection to the hybrid network, allowing redirection of SIP requests to SIP URIs, PSTN endpoints, or to a combination of both.

Typically, the call flow is as follows:

1. Upon notification of an incoming PSTN call, the Gateway establishes an outbound SIP INVITE request based on the routing rules.
2. If this request is sent toward a redirect SIP server, that SIP server will respond with a 3XX type response (such as ‘302 Moved temporarily’) with one or many new URIs in the ‘Contact’ header (and potentially specifying a different q-value for each of these URIs).
3. The Gateway then tries to establish the outbound SIP leg according to that response. The *Netborder.gw.maxRoutingRulesMatches* global configuration property (in the *gw.properties* file) determines the maximum number of endpoints that may be contacted to establish the call. A new contact is always compared with previous ones to guard against infinite loops.

The following diagram illustrates the SIP message call flow for a redirection attempt to two different URIs, where the first one is ‘Busy’, and the second accepts the call.



Note that the Gateway expects a 3XX response prior to the SIP target sending a '180 Ringing'.

Once the Gateway receives a 180 message, it goes into the 'Ringing' state, and answers the PSTN incoming call (at which point the switch typically generates ringback to the user). Once in the 'ringing' state, the Gateway will refuse 3XX messages, since to do otherwise could impede performance and quality (such as the user hearing ringback from the switch, then busy from the new SIP target).

For specific redirect platforms that do generate a '180 Ringing' prior to sending a 3XX, this problem can be circumvented by forcing the Gateway to ignore 180 messages on that SIP leg. This is accomplished via the routing rules, by setting the parameter *sip.out.ignore180* to 'true' in the rule that sends *the original SIP INVITE to the redirect server*. This way, the 180 message from the redirect server will be ignored, the Gateway will never go into the 'ringing' state and will therefore accept a 3XX message.

It should be noted, however, that most redirect servers do not send '180 Ringing' responses and instead send the 3XX response immediately after the '100 Trying', as depicted in the example.

Below is the routing rule used by the Gateway to establish the content of the SIP INVITE requests following a redirect primitive. **This rule should not be edited since the format of SIP messages following a redirect primitive is strictly dictated by the SIP standard.**

```
< rule name =“ Redirect_to_sip "
      outbound_interface =“ sip " qvalue =" 0.1 ">

  <!–- If this is a redirect, this will match and put the contact header in %0  -->
  < condition param =“ sip.in.redirect.Contact " expr =“ (.*) " />

   < out_leg name =“ default " media_type =“ sendrecv “ >
     <!-- Set the CallerID in the ‘From’ URI -->
      < param name =“ sip.out.redirect.Contact " expr =“ sip:%0 "/>
   </ out_leg >
</ rule >
```

Calls can also be redirected to a PSTN endpoint. In this case, a SIP call is never established, and the two PSTN call legs are bridged on the Gateway, which in effect acts as a PSTN switch.

The following rule can be used/augmented/modified to trigger a PSTN-redirect. If the telephony board in the Gateway is equipped with a CTBus, the two call legs are connected together on the bus, thereby minimizing host resources and latency.

```
< rule name =“ Redirect_to_pstn " outbound_interface =“ pstn " qvalue =" 0.2 ">

  <!–- If this is a redirect, this will match and put the target phone number in %0  -->
  < condition param =“ sip.in.redirect.Contact "
              expr =“^Contact: sip: ([0-9]+) @GW_HOST_IP:GW_SIP_PORT "/>

   < out_leg name =“ default " media_type =“ sendrecv “ >
     <!-- Set the outbound phone number and outbound device group -->
```

```
    < param name =“ pstn.out.phoneNumber " expr =“ %0 "/>
    < param name =“ pstn.out.deviceGroup " expr =“default"/>
  </ out_leg >
</ rule >
```

SIP transport: UDP/TCP

The transport used to carry SIP messages can be set on a per-call basis by using the *sip.out.transport* parameter in the routing rules. The default value of this parameter is *UDP*.

Below is an example of a rule that uses *TCP (Transmission Control Protocol)*, rather than *UDP (User Data Protocol).*

```
< out_leg name =“ Voice_Platform " media_type =“ sendrecv “ >
    <!-- Set the CallerID in the ‘From’ URI -->
    < param name =“ sip.out.from.uri "
                expr =“ sip:%0@GW_HOST_IP:GW_SIP_PORT "/>
    < param name =“ sip.out.from.displayName " expr =“ Gateway "/>

    <!-- Set the outbound URIs to point to URI of the form sip:5551212@acme.com -->
    < param name =“ sip.out.requestUri " expr =“ sip:%1@acme.com “ />
    < param name =“ sip.out.to.uri " expr =“ sip:%1@acme.com "/>
    <param name =“ sip.out.transport " expr=“ tcp "/>
    < param name =“ sip.out.to.displayName " expr =“ App "/>
  </ out_leg >
```

To use TCP, make sure the Gateway is listening to a TCP port by using the *Netborder.sip.userAgent.IPAddress* parameter.

Early media

Early Media is the ability of two SIP 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.

This functionality is often implemented for user agents that provide custom ringback, or are performing Third Party Call Control and will later transfer the call to another party without connecting the call at all. Typically, the user agent sends a 1XX provisional response (for example, ‘180 Ringing’ or ‘183 Session Progress’) and begins playing the audio stream prior to receiving an acknowledgment (or ‘200 OK’ message).

The Gateway provides the required signaling by adding a Progress Indicator (PI) Information Element (IE) to an ISDN message based on the call state of the event. Specifically, it consists of adding a PI saying that some in-band information is available (PI value=8) to an ALERT or PROGRESS ISDN message.

As there are some PSTN protocols that do not support signaling of early media (like CAS or Analog), this functionality can be disabled using the parameter *Netborder.sip.out.acceptEarlyMedia*. In the *gw.properties* file, simply set the value of this parameter to “false”.

Similarly, it is possible to change the behaviour of the Gateway to make early media function even in a call flow not supported by ISDN. For example, a SIP user agent that sends a 180 response message without Session Description Protocol (SDP) and then a 183 response with SDP. This is accomplished by two routing rule parameters, *sip.out.ignore180* and *sip.out.accept183*. For more information, see the Outbound Routing Rule Parameters Table.

SIP-initiated calls

The SIP application may also initiate outbound calls to the PSTN. According to the default routing rules, an INVITE message is sent to the Gateway with a 'Request URI' field adhering to the following format:

- `sip:<phone number>@<gateway-host>:<gateway-ua-port>`

  where:

  - `<phone number>` is the destination phone number as it would be dialed on the telephone line used by the Gateway
  - `<gateway-host>` is the name or IP address of the Gateway host
  - `<gateway-ua-port>` is the port used by the Gateway's SIP User Agent.

For example, if the Gateway system runs on a host named 'gateway' and the phone number to be dialed is 514-555-1212, then the Request URI of the INVITE message would be set to "*sip:5145551212@gateway:5066*".

NOTE: Ensure that all the phone numbers map to the appropriate dialing plan.

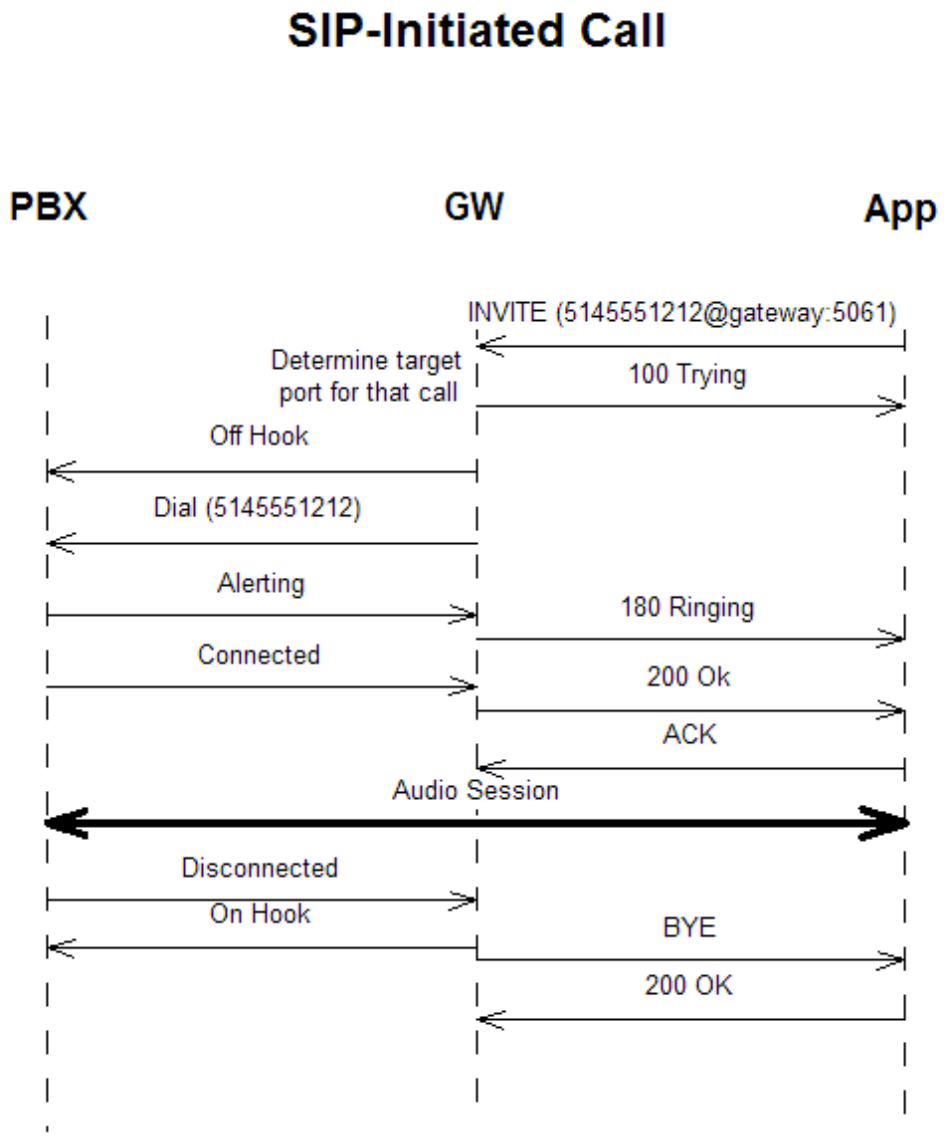
The call flow should resemble the following:

1. Upon reception of the INVITE, the Gateway will try to reserve the most appropriate PSTN resource to perform the outbound call based on the routing rules and the resources' availability.

   If no resource is available, a message '480 Temporarily not available' is returned. If a resource is available, then the call is initiated.

2. Upon receiving the 'Alerting' message from the switch, the Gateway sends a '180 Ringing' to the originating SIP application.
3. On reception of the 'Connected' signal on the telephony line, the call is formally established with the SIP endpoint and the end-to-end audio path enabled.
4. Upon detecting a hang-up on either side, the Gateway disconnects the other leg.

The following diagram illustrates the interactions between the parties.



In some third party call control scenarios (RFC 3725), it is possible for the SIP application to invite the Gateway into a session without offering any SDP (Session Description Protocol). In this case, the Gateway will try to establish the PSTN call and offer its own SDP in the '200 OK' message.

The Gateway will expect to receive the RTP endpoint information in the ACK message back from the SIP application. If the ACK message does not contain the RTP endpoint information, the Gateway will have no choice but to immediately terminate the session using the BYE message.

Early media

When calling an outside number on the PSTN there are conditions—mainly using ISDN—under which some messages are played in-band. For instance, if the Gateway detects a busy tone, it will report the correct corresponding code to the calling SIP user Agent (with a '486 Busy' message).

However, there are occasions when the tone is not recognized (for example, a custom tone), or the callee wants to play a custom ring or even wishes to interact with the caller before being connecting (for example, in the case of 800 numbers, which wait for a special DTMF before transferring to an agent and connecting in order to confirm call rate or IVR). In these cases, the Gateway must play the audio sent by the callee, and notify the caller that there is audio to be heard (actually the audio streams are started in Receipt [Rx] and Transmit [Tx] as the caller may need to interact). This feature is called Early Media and is possible in SIP thanks to the '183 Session Progress' message.

For information on configuring this feature, please see Outbound call propertiesOutbound call properties on page 170. The default value of the parameter is 'as-needed'.

Here is an example of a routing rule in which this feature is disabled.

```
<rule name="default_pstn_out" outbound_interface="pstn" qvalue="0.001">
  <condition param="sip.in.requestUri.canonical"
    expr="sip:([0-9]+)@GW_HOST_IP:GW_SIP_PORT"/>
  <out_leg name="" media_type="sendrecv">
    <param name="pstn.out.phoneNumber" expr="%0"/>
    <param name="pstn.out.deviceGroup" expr="default"/>
    <param name="pstn.out.cpa.enable" expr="false"/>
    <param name="pstn.out.earlyMediaMode" expr="never"/>
  </out_leg>
</rule>
```

A value of *"never"* for the *pstn.out.earlyMediaMode* parameter will disable early media for all configurations, under all circumstances.

You may encounter incompatibilities with some SIP user agents that do not support SIP 180 responses with an SDP in the body, or do not implement the '183 Session Progress' message. In such cases, the following two global parameters may be of help:

- Netborder.*sip.in.doNotSend180After183;* and,
- Netborder.*sip.in.override1XXWithSDP*.

See the Global Configuration Properties Table in Appendix B for more information on these parameters.

SIP status codes

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.
- **4XX, 5XX, and 6XX:** Indicates failure in processing of the SIP request.

Telephony error codes

The following list contains the most frequent traditional telephony error codes you will encounter while using the Gateway. These codes represent the Gateway's mapping of telephony error conditions (which occur mainly on SIP-originated calls or SIP-initiated transfers) into standard SIP responses.

Telephony Error Codes		
Response Code	***Response Name***	***Explanation***
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 as specified on the outbound telephony lines. **Recommended Action:** Verify the phone number and its format.
408	Request Timeout	The Gateway failed to establish a connection within the specified timeframe (no answer from remote party). **Recommended Action:** Call again later.

Telephony Error Codes		
Response Code	***Response Name***	***Explanation***
480	Temporarily Unavailable	No telephony resources matching the appropriate routing rules could be acquired for the outbound call leg. **Recommended Action:** Verify the provisioning of outbound lines, and/or try again later. Ensure there is an actual telephony line available that matches the resource descriptions in the routing rules.
486	Busy Here	Indicates a busy signal on the outbound telephone number. **Recommended Action:** Try again later.
488	Not Acceptable Here	No routing rule matched the specific request made by the SIP application. **Recommended Action:** Review routing rules and modify them to match the specific case requested.
5XX Server-Error		
500	Internal Server Error	An internal error has occurred. **Recommended Action:** Collect detailed logs and contact Technical Support.
503	Service Unavailable	An error occurred at the physical level. Possible causes: the telephony board is not responding, or the physical line is unplugged. **Recommended Action:** Verify that the telephony hardware is functioning properly.
504	Server Timeout	The Gateway failed to receive an 'alerting' message from the switch within the allowable timeframe. **Recommended Action:** Call again later.
6XX Global Failure		
603	Decline	The PSTN user dropped the call. Typically, this happens when using advanced call progress analysis and the callee hangs up before media detection is complete.

ISDN to SIP mapping

In ISDN, signaling is richer than in Analog/CAS. Information must be mapped to SIP signaling in order to precisely report the different events. These events could be SIP/ISDN messages (for example, an ALERT message, which is similar to the SIP 180 provisional response), or could even form part of SIP/ISDN messages. In particular, the cause located in a 'Cause code' Information Element (IE) is strongly linked with the SIP status code in a final SIP response.

For the definition of the different ISDN Cause values and SIP status codes, see the reference documents ITU Q.850 and RFC 3261, respectively.

Note that not all cause codes could be mapped to a status code and result at the call end. The mapping operated by the Gateway is based on RFC 3398 "*Integrated Services Digital Network (ISDN) User Part (ISUP) to Session Initiation Protocol (SIP) Mapping*".

The mapping between an ISDN cause code IE and SIP status code is detailed in the table below. All unreferenced values (ISDN cause values range from 0 to 127) are mapped to the SIP status code '500 Internal server error'.

Mapping of ISDN Cause Codes to SIP Status Codes	
ISDN Cause Code	***SIP Status Code***
1 Unallocated number	404 Not found
2 No route to network	404 Not found
3 No route to destination	404 Not found
17 User busy	486 Busy here
18 No user responding	408 Request timeout
19 No answer from the user	480 Temporarily unavailable
21 Call rejected	403 Forbidden
22 Number changed	410 Gone
26 Non-selected user clearing	404 Not found
27 Destination out of order	502 Bad gateway
28 Address incomplete	484 Address incomplete
29 Facility rejected	501 Not implemented
31 Normal unspecified	480 Temporarily unavailable
34 No circuit available	503 Service unavailable
38 Network out of order	503 Service unavailable
41 Temporary failure	503 Service unavailable
42 Switching equipment congestion	503 Service unavailable
47 Resource unavailable	503 Service unavailable

Mapping of ISDN Cause Codes to SIP Status Codes	
ISDN Cause Code	**SIP Status Code**
57 Bearer capability not authorized	403 Forbidden
58 Bearer capability not presently available	503 Service unavailable
63 Service not available	503 Service unavailable
65 Bearer capability not implemented	488 Not Acceptable here
70 Only restricted digital avail	488 Not Acceptable here
79 Service or option not implemented	501 Not implemented
88 Incompatible destination	503 Service unavailable
95 Invalid message	400 Bad request
102 Recovery of timer expiry	504 Server time-out
111 Protocol error	500 Internal server error
127 Interworking unspecified	500 Internal server error
Others	500 Internal server error

SIP to ISDN mapping

The inverse mapping is referenced in the table below, from SIP status code to ISDN cause code.

Mapping of SIP Status Codes to ISDN Cause Codes	
SIP Status Code	**ISDN Cause Code**
400 Bad request	41 Temporary failure
401 Unauthorized	21 Call rejected
402 Payment required	21 Call rejected
403 Forbidden	21 Call rejected
404 Not found	1 Unallocated number
405 Method not allowed	63 Service not available
406 Not acceptable	79 Service or option not implemented
407 Proxy authentication required	21 Call rejected
408 Request timeout	102 Recovery of timer expiry
409 Conflict	41 Temporary failure
410 Gone	22 Number changed
411 Length required	127 Interworking unspecified
413 Request Entity too long	127 Interworking unspecified
414 Request-URI too long	127 Interworking unspecified
415 Unsupported media type	79 Service or option not implemented
416 Unsupported URI Scheme	127 Interworking unspecified
420 Bad extension	127 Interworking unspecified

Mapping of SIP Status Codes to ISDN Cause Codes	
SIP Status Code	***ISDN Cause Code***
421 Extension Required	127 Interworking unspecified
423 Interval Too Brief	127 Interworking unspecified
480 Temporarily unavailable	18 No user responding
481 Call/Transaction Does not Exist	41 Temporary failure
482 Loop Detected	127 Interworking unspecified
483 Too many hops	127 Interworking unspecified
484 Address incomplete	28 Address incomplete
485 Ambiguous	1 Unallocated number
486 Busy here	17 User busy
487 Request Terminated	127 Interworking unspecified
488 Not Acceptable here	127 Interworking unspecified
500 Internal server error	41 Temporary failure
501 Not implemented	79 Service or option not implemented
502 Bad gateway	38 Network out of order
503 Service unavailable	41 Temporary failure
504 Server time-out	102 Recovery of timer expiry
505 Version not implemented	127 Interworking unspecified
513 Message Too Large	127 Interworking unspecified
580 Precondition Failed	47 Resource unavailable
600 Busy everywhere	17 User busy
603 Decline	21 Call rejected
604 Does not exist anywhere	1 Unallocated number

Chapter 7: Routing Rules

The routing rules are a set of rules that are applied when a call is received, and help determine how the resulting outbound call is handled.

The routing rules file can be used, together with the PSTN configuration file (see Chapter 5), to define the physical resources used by the Gateway. For example, you might create an element in the PSTN configuration file and then use that element in the routing rules file to write flexible rules that determine how to route calls based on a number of factors.

This chapter contains the following topics:

What are routing rules?

When an incoming call is received, the routing rules are fetched by the Gateway to a web server. The rules obtained, which are based on the characteristics of the call, are then applied to the incoming call in order to find an outbound location. A routing rule is, by definition, a logical expression that processes an incoming call (regardless of whether it originates from the PSTN or the IP network) and, once the rules' conditions are matched, produces the parameters for the associated outbound call.

The routing rules file is located here:

- *[GATEWAY_HOME]\config\routing-rules.xml*

  where *[GATEWAY_HOME]* is the root folder of the installation (for example, C:\Program Files\Netborder\Express\Gateway*config\routing-rules.xml*).

The location of the routing rules file is specified in the *gw_sangoma.properties* file by the *Netborder.gw.routingRulesUrl* parameter. Since the Gateway embeds a web server, the routing rules are by default obtained from the Gateway web server. However, by modifying the URL location property, any web server can be used.

Out of the box, the Routing Engine is configured to route all incoming PSTN calls to the SIP user agent at the URI specified at installation time. It is also configured to support the calling properties as described in the preceding chapter.

If you are running your target application (either a SIP application or a SIP proxy) at the location specified at installation, and you are using the basic Gateway functionality, **you do not need to modify the routing configuration**.

However, the routing rules file provides powerful customization capabilities, which you may wish to explore.

Modifying routing rules

The Routing Engine provides application developers with the ability to customize the way calls are routed.

The routing rules file can be modified manually in one of two ways:

- By using a text or XML editor of your choice and modifying the *[GATEWAY_HOME]\config\routing-rules.xml* file (where *[GATEWAY_HOME]* is the root folder of the installation). The Gateway must be restarted for changes to take effect.
- By using the Gateway Web Interface and clicking on the **Routing Rules** tab at the top of the page (for more information on the routing rule editing web page, see Editing the routing rules on page 93). Once the file modifications have been saved to disk, the Gateway fetches the new rules on the next incoming call and uses them to make routing decisions.

Routing Engine

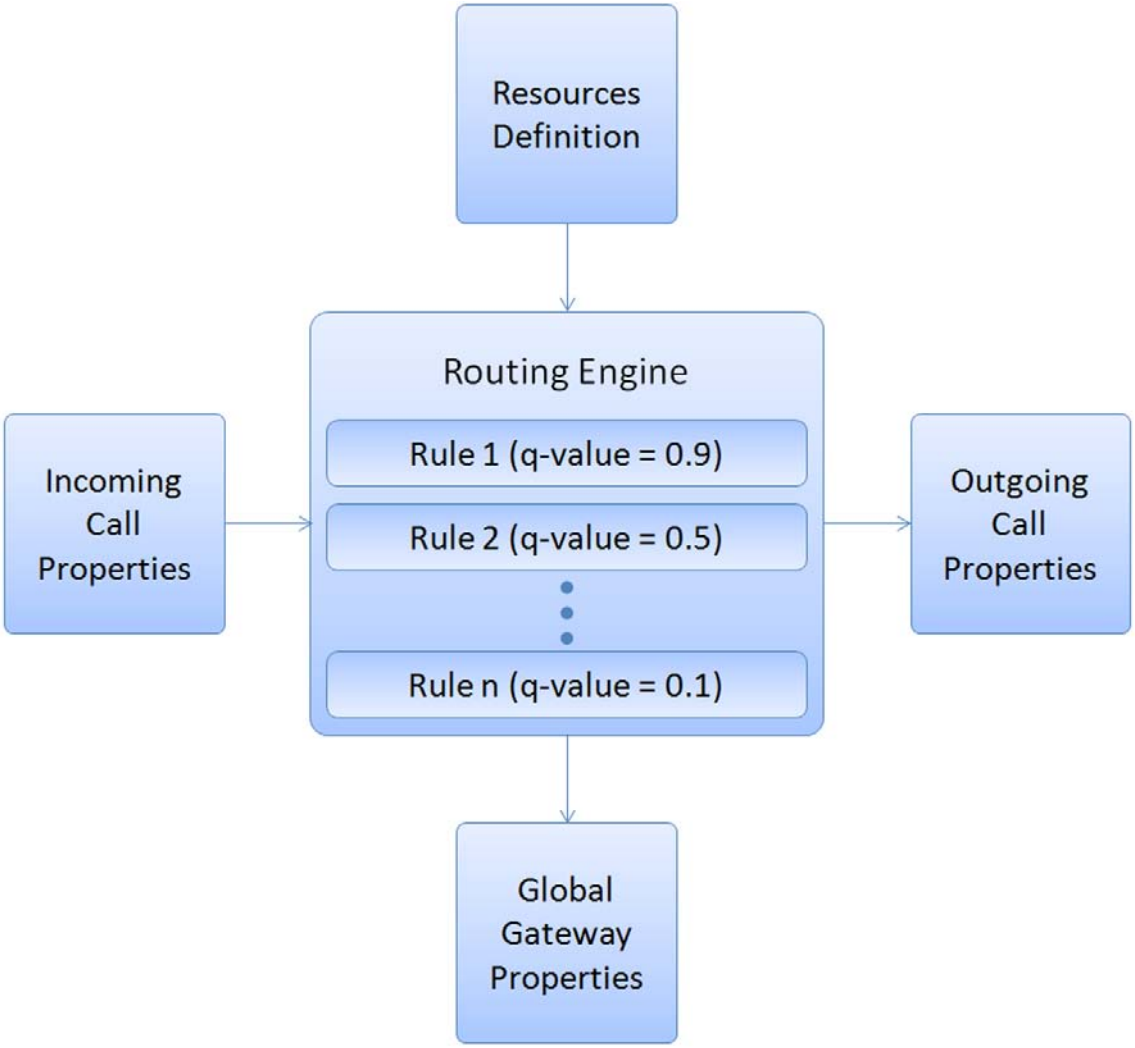
When the Routing Engine receives a routing request with some incoming call properties, it does the following:

- fetches the routing rules
- applies the routing rules to the request based on the resource definitions and general Gateway properties (such as the Gateway's host name)
- produces the outgoing call properties.

The call is then relayed to a destination, either PSTN or VoIP, by matching the properties produced by the Routing Engine.

If no routing rule matches the incoming request, the incoming call is rejected with an appropriate SIP response code (see Telephony error codes on page 157). If the outbound destination is VoIP, multiple call legs can be established using the same incoming PSTN call.

The following diagram depicts at a high level the operation of the Routing Engine.



Q-value

Although multiple rules may be active at the same time, not all rules have the same priority. A rule's priority level is defined by its *q-value*. The q-value of a rule is a floating-point number between 0 and 1, with 1 representing the highest priority.

The term "q-value" comes directly from the SIP standard. In SIP, an application can register to a SIP proxy and specify a q-value that represents its availability (floating-point number between 0 and 1).

When you create a new rule, you must assign it a q-value. The Routing Engine processes routing requests by applying the entire set of routing rules, sorted in descending order of q-value. The first rule to trigger on the incoming request's properties will produce the outbound call properties.

If an outbound call cannot be established using the highest priority rule that is triggered, then the Routing Engine will try to establish a call using a lower priority rule that meets all of the conditions (if any), until the maximum value is reached (specified by parameter *Netborder.gw.maxRoutingRulesMatches*).

CAUTION: Assign q-values carefully. If two rules with an equal q-value trigger a request, the Routing Engine cannot guarantee or predict which one will be used at runtime.

Using definitions and properties in routing rules

When creating a routing rule, the following types of variables can be used to process the incoming request:

- Global gateway properties
- Resource definitions
- Inbound call properties.

Global Gateway properties

There are a few global Gateway properties that can and should be used in the routing rules for this release. They include the following:

- **GW_SIP_PORT:** The port value of the first IPAddress in the Netborder.*sip.UserAgent.IPAddress* property list.
- **GW_HOST_IP:** Local host IP address.
- **GW_CALL_ID:** The unique call identifier generated by the Gateway for each PSTN-to-SIP call.

Resource definitions

Resource definitions are used to define the outbound call properties. For example, a rule monitoring SIP-initiated calls could direct the call to a specific device group. A rule references a specific resources group by the ID property of the *<resourcesGroup>* element (see Creating and configuring a resources group on page 135). For example, a set of rules could route in priority SIP calls to a priority outbound pool or to a backup outbound pool depending on the type of SIP application requesting the call from the PSTN. A rule can target a specific device group by setting the outbound call property pstn.out.resourcesGroup to the appropriate group name or ID.

Inbound call properties

Finally, the routing rules can use numerous properties of the incoming call to determine the outbound properties. A call coming into the Gateway is either a PSTN-originated call or a SIP-initiated call. A PSTN-originated call will have non-nil values in the inbound properties *pstn.in.** and nil values in *sip.in.**. For a SIP-initiated call, the opposite is true: it will have nil values in the inbound properties *pstn.in.** and non-nil values in *sip.in.**.

The following table lists the properties that can be accessed from the routing rules.

Inbound Routing Rule Parameters	
Parameter	***Description***
pstn.in.channelName	The Sangoma channel name on which the PSTN call came in. This would be a string in the following format: • `S<span index>C<channel index>` where `<span index>` is the number of the span, and `<channel index>` is the number of the channel.
pstn.in.dnis	Dialled number. Available only in digital configurations.
pstn.in.ani	Calling number. In analog configurations, this is the CallerID. If not available, this property is set to "*unknown*".
pstn.in.isdn.setup.iiDigits	Automatic Number Identification (ANI) II digits in the ISDN setup message. These digits identify the type of originating station. Only available with ISDN signalling.
pstn.in.isdn.setup.ie.0xZZ.0xYY	The content of ISDN Information Element with codeset **0xZZ** and identifier **0xYY**, in hexadecimal format. All ISDN Information Elements (IE) of the SETUP message can be accessed using this property. The IE content is stored in an hexadecimal format where each octet is separated by a space, **without** the '0x' prefix.

Inbound Routing Rule Parameters	
Parameter	***Description***
sip.in.requestUri sip.in.requestUri.canonical	The URI to which the call request was made (and its canonical version).
sip.in.from.uri sip.in.from.uri.canonical	The caller's URI (and its canonical version).
sip.in.from.displayName	The caller's display name if available.
sip.in.to.uri sip.in.to.uri.canonical	The callee's URI (and its canonical version).
sip.in.to.displayName	The callee's display name if available.
sip.in.referTo sip.in.referTo.canonical	The address in the "Refer To" header of a REFER message (and its canonical version).
sip.in.referredBy.uri sip.in.referredBy.uri.canonica	The URI of the application requesting the transfer (and its canonical version).
sip.in.referredBy.displayName	The display name of the SIP application requesting the transfer.
sip.in.redirect.Contact	The content of the contact header on reception of a SIP redirect (3XX) message.
sip.in.header.Via	The content of the Via header.
sip.in.header.CSeq	The content of the CSeq header.
sip.in.header.Call-ID	The content of the Call-ID header.
sip.in.header.Content-Length	The content of the Content-Length header.
sip.in.header.Contact	The content of the Contact header.
sip.in.header.HeaderName	For acquiring the content of an arbitrary header in the incoming SIP message (incoming call or transfer request). To obtain the value of *MyPrivateHeader*, you would use *sip.header.MyPrivateHeader*. Refer to the *Release Notes* for limitations.

NOTE: Canonical versions of the URI parameters are fully expanded of the form *sip:something@hostname:portnumber*. Note that 'sip' must be in lowercase, and that the hostname is expanded to dotted IP notation. For example: *sip:1026@192.168.11.207:5066*.

Outbound call properties

The purpose of a routing rule is to set the properties of an outgoing call (either PSTN or SIP) based on the information gathered about the incoming call, as well as global properties and resource data.

The following table lists the outbound properties that can be set in a routing rule using the *<out_param>* element:

Outbound Routing Rule Parameters	
Parameter	***Description***
pstn.out.phoneNumber	For setting the phone number as it should be dialed on the telephony channel.
pstn.out.ani	For setting the ANI (CAS, default value is 5678), or the calling number information (ISDN).
pstn.out.deviceGroup	For setting the resources group to select in making the outbound call.
pstn.out.isdn.earlyConnect Mode	For setting the conditions, if any, under which the Gateway can connect the call, even before receiving the PRI CONNECTED message. Valid values are: 'never', 'always' or 'as-needed'. Default value is 'never'. The 'always' setting will declare connected on all PROGRESS messages (that is, as soon as the Gateway gets ringback from the PSTN). The 'as-needed' setting will declare connected only upon reception of a PROGRESS message with no cause IE or a NORMAL cause IE.
pstn.out.earlyMediaMode	For setting the conditions, if any, under which the Gateway will start the media streams and notify the calling SIP *UA (User Agent)* of an '183 Session Progress' (message that there is some audio prior to call connection that the caller should hear). Valid values are: 'never', 'always' or 'as-needed' (default).
pstn.out.isdn.setup.iiDigits	For setting the automatic Number Identification (ANI) II digits sent with the SETUP message. These digits identify the type of originating station. Only available with ISDN signalling.
pstn.out.isdn.setup.ie.0xZZ.0xYY	For setting the content of ISDN Information Element with codeset **0xZZ** and identifier **0xYY**. Any ISDN Information Elements (IE) of the SETUP message can be set using this property. The IE content is specified in a hexadecimal format where each octet is separated by a space, **without** '0x' prefix. Any IE defined here will be added to the SETUP message **without validation** of its

Outbound Routing Rule Parameters	
Parameter	**Description**
	content, overriding any automatically generated IE. Use this feature with care as it may send corrupted message without warning being generated by the application. The IE will be correctly placed in the message according to Q931 specification and many IE may be defined in the same message.
sip.out.ignore180	Can be true or false (default is false). When set to true, it will ignore the 180 messages from the target SIP application and not go in the 'ringing' state. Can be useful if a SIP user agent sends a 180 without SDP followed by a 183 with SDP, as this situation cannot be reproduced in ISDN, and the Gateway will fail to establish early media.
sip.out.accept183	Can be true or false (default is false). When set to true, it will accept a 183 message as a 180 and go in the 'ringing' state. Can be useful in the same situation as the *sip.out.ignore180* property.
sip.out.requestUri	For setting the URI to which the call request is going to be made.
sip.out.from.uri	For setting the caller's URI. Typically used to indicate the the ANI or CallerID of an incoming PSTN call, if available.
sip.out.from.displayName	For setting the caller's display name if available.
sip.out.to.uri	For setting the callee's URI.
sip.out.to.displayName	For setting the callee's display name.
sip.out.redirect.Contact	For setting the contact header following a redirect primitive.
sip.out.header.HeaderName	For setting arbitrary headers in the outgoing INVITE message. For example, to set *MyPrivateHeader* to value 'foo', you would use *sip.out.header.MyPrivateHeader=foo*. Refer to the *Release Notes* for limitations.

Now that we have identified the sources of input and output, we will see in the next section how to pull all of this information together into a comprehensive routing rule.

Routing rule constructs

Routing rules are described using an XML format. The following table lists the elements that are used to define a rule.

Routing Rule Elements		
Element	***Description***	***Attributes***
<rule>	The <rule> element is used for each of the system's routing rules	• **name** (string): The name of the rule. • **outbound_interface** (string): This is where the call should be directed (outbound direction). Can be set to either 'pstn' or sip'. • **qvalue** (float): Priority of the rule between 0 and 1, 1 being the highest priority.
<condition>	This element is used to test one condition on a number of properties. Multiple <condition> elements or "tags" are permitted within one rule; however, all conditions must be met for that rule to trigger.	• **param** (string): The name of the parameter on which the condition is tested. Names of suitable parameters are provided in the Inbound Routing Rule Parameters Table and the Outbound Routing Rule Parameters Table. • **expr** (string): Regular expression. Test submitted to the input parameter. The input parameter must successfully pass the test for the condition to be met. • **except** (string): Regular expression. Negative test to be performed on the input parameter. If the 'except' property is used, then the 'expr' must return true and the 'except' must return false for the condition to be met.
<out_leg>	An <out_leg> element is used for each outbound call leg resulting from a triggered rule on incoming calls.	• **name** (string): The name of the out leg. • **media_type** (string): The type of media connection. Can only be set to 'sendrecv' or 'recording' in this release. • **outbound_interface** (string): Can be set to 'sip' or 'pstn' to overwrite the outbound interface of the rule, for that <out_leg> only.

Routing Rule Elements		
Element	***Description***	***Attributes***
<param>	This element is used to set an outbound call leg property. One outbound parameter is set per <param> element or "tag". Multiple tags are allowed within a single out leg.	• **name** (string): The name of the outbound call property to be set. Names of suitable parameters are provided in the Outbound Routing Rule Parameters Table. • **expr** (string): Value to set in the outbound call property. May contain the results of regular expressions and/or inbound and general properties.

The routing rule syntax leverages the full power of a regular expression processor, which is integrated into the Routing Engine. A regular expression is a string that is used to describe or match a set of strings, according to certain syntax rules. If you are not familiar with regular expressions, we encourage you to consult widely available literature on this subject, including the following tutorial at Perldoc.com: http://www.ayni.com/perldoc/perl5.8.0/pod/perlretut.html.

The specific regular expression package used by the Gateway is documented at the following location: http://www.pcre.org/pcre.txt.

The regular expressions in routing rules are most often used to perform pattern matching on the input parameters and/or to format the outbound properties. An atom within a regular expression is delimited within the rule using parentheses (), and can be called later in the rule by using the percent sign %. The first atom in a rule can be accessed via variable %0, the second one via variable %1, and so on. A regular expression only lives within the scope of the <*rule*> element within which it is defined. See the following routing rules examples for clarification.

Routing rule examples

Below are a four examples of how you might use routing rules to accomplish specific ends.

Example 1: Incoming SIP call dialing a 3- or 4-digit extension on PBX

Here is an example of a single rule that triggers when an incoming SIP call has the following format: `sip:<3-4-digit-ext>@<gateway-host>:<gateway-ua-port>`, where:

- `<3-4-digit-ext>` is a three or four-digit extension to dial on the PBX for the outbound call leg
- `<gateway-host>` is the name or IP address of the Gateway host
- `<gateway-ua-port>` is the port number of the Gateway's SIP User Agent (5066 by default).

```
<!-- In SIP, out PSTN, dialing a 3 or 4 digit extension on the PBX -->
< rule name =" outbound_PSTN_extension " outbound_interface =" pstn " qvalue =" 0.1 ">
<!--
 Look if the incoming URI is made of 3 or 4 digits followed by the gateway host (global GW_HOST_IP)
and user agent port (GW_SIP_PORT)
 -->
 < condition param =" sip.in.requestUri.canonical " expr = " ([0-9]
{3,4})@GW_HOST_IP:GW_SIP_PORT " />
<!--
 The text found by regular expression ([0-9]{3,4}) can be accessed via variable %0. Set the outbound
phone number to dial
 -->
 < out_leg name =“ default " media_type =“ sendrecv “ >
          < param name =" pstn.out.phoneNumber " expr =" %0 "/>
   </ out_leg >
/>

</ rule >
```

Example 2: Incoming SIP call dialing an external number

Here is an the example of another rule that performs exactly the same function as the one above, except that it dials a '9' followed by a pause (represented by a comma in telephony), followed by an external phone number.

```
<!-- In SIP, out PSTN, dialing an external number out of the PBX -->
< rule name =" outbound_PSTN_external " outbound_interface =" pstn " qvalue =" 0.1 ">
<!--
 Look if the incoming URI is made of 7 digits followed by the gateway host (global GW_HOST_IP) and
user agent port (GW_SIP_PORT)
                        -->
  < condition param =" sip.in.requestUri.canonical "
           expr =" ([0-9]{7,})@GW_HOST_IP:GW_SIP_PORT " />
<!-- Set the outbound phone number to dial "9," + phone number -->
  < out_leg name =“ default " media_type =“ sendrecv “ >
< param name =" pstn.out.phoneNumber " expr =" 9,%0 " />
        </ out_leg >
</ rule >
```

The same type of rule could be used to detect a long distance number and handle it appropriately on the PBX line.

Example 3: DNIS-based routing

Routing rules allow you to select a target SIP destination based on the properties of the incoming call. With analog lines, the number of properties for the incoming call is minimal (with a digital configuration, on the other hand, much more information is known about an incoming PSTN call). Still, you might find it useful, for example, to route all of the incoming calls on a particular port range to a specific application, and calls coming into other ports to another SIP destination.

Here is an example of a rule that could be used to direct the received calls on the first span with different numbers (DNIS) to different SIP URIs.

```
<!– DNIS-based Routing -->

< rule name =“ DNIS_Routing " outbound_interface =“ sip " qvalue =" 0.1 ">

  <!-- Look if the incoming PSTN call is on the first Strong_20_Emphasis -->

< condition param =“ pstn.in.channelName " expr =" B1T.* " />

  <!–- Retrieve and store the ANI in variable %0  -->
  < condition param =“ pstn.in.ani " expr =“ (.*) " />

 <!–- Retrieve and store the DNIS in variable %1 -->
  < condition param =“ pstn.in.dnis " expr =" ([0-9]*) " />

   < out_leg name =“ default " media_type =“ sendrecv “ >
     <!-- Set the CallerID in the ‘From’ URI -->
      < param name =“ sip.out.from.uri "
                    expr =“ sip:%0@GW_HOST_IP:GW_SIP_PORT "/>
      < param name =“ sip.out.from.displayName " expr =“ Gateway "/>

      <!-- Set the outbound URIs to point to URI of the form sip:5551212@acme.com -->
      < param name =“ sip.out.requestUri " expr =“ sip:%1@acme.com “ />
      < param name =“ sip.out.to.uri " expr =“ sip:%1@acme.com "/>
      < param name =“ sip.out.to.displayName " expr =“ App "/>
   </ out_leg >

</ rule >
```

Example 4: Application failover using q-value

As discussed earlier (Q-value on page 166), when multiple rules are triggered, the Gateway will apply the rule with the highest priority (q-value). Multiple SIP destinations with different priorities could thus be targeted for a single PSTN call. This way, if the Gateway is not able to connect with the highest priority destination, it will try the second highest priority, and so on, according to the maximum number of rules that can be used to establish a call (as per the *Netborder.gw.maxRoutingRulesMatches* global configuration property).

Below are sample routing rules that could be used to provide application failover. In this example, the address *sip:primary@acme.com* is targeted in priority (qvalue is set to 0.2). If this application does not respond, then the second rule (qvalue set to 0.1) will be executed and attempt to contact ‘*sip:backup@acme.com*’.

```
 < rule name =“ Primary_Routing " outbound_interface =“ sip " qvalue =" 0.2 ">

  <!–- Retrieve and store the ANI in variable %0  -->
  < condition param =“ pstn.in.ani " expr =“ (.*) "/>

 <!–- Retrieve and store the DNIS in variable %1 -->
  < condition param =“ pstn.in.dnis " expr =" ([0-9]*) "/>

   < out_leg name =“ default " media_type =“ sendrecv “ >
     <!-- Set the ANI in the ‘From’ URI -->
      < param name =“ sip.out.from.uri "
                  expr =“ sip:%0@GW_HOST_IP:GW_SIP_PORT "/>
      < param name =“ sip.out.from.displayName " expr =“ Gateway "/>

      <!-- Set the outbound URIs to point to URI of the primary target -->
      < param name =“ sip.out.requestUri " expr =“ sip:primary@acme.com “ />
      < param name =“ sip.out.to.uri " expr =“ %1 "/>
      < param name =“ sip.out.to.displayName " expr =“ App "/>
   </ out_leg >

</ rule >

< rule name =“ Backup_Routing " outbound_interface =“ sip " qvalue =" 0.1 ">

   <!–- Retrieve and store the ANI in variable %0  -->

   < condition param =“ pstn.in.ani " expr =“ (.*) " />
   <!–- Retrieve and store the DNIS in variable %1 -->
   < condition param =“ pstn.in.dnis " expr =" ([0-9]*) " />

    < out_leg name =“ default " media_type =“ sendrecv “ >
      <!-- Set the ANI in the ‘From’ URI -->
      < param name =“ sip.out.from.uri "
                  expr =“ sip:%0@GW_HOST_IP:GW_SIP_PORT "/>
      < param name =“ sip.out.from.displayName " expr =“ Gateway "/>

      <!-- Set the outbound URIs to point to URI of the backup target -->
      < param name =“ sip.out.requestUri " expr =“ sip:backup@acme.com “ />
      < param name =“ sip.out.to.uri " expr =“ %1 "/
      < param name =“ sip.out.to.displayName " expr =“ Backup App "/>
    </ out_leg >

</ rule >
```

Default routing rules

The default routing configuration file ships with a number of pre-configured routing rules. To customize the Gateway's routing behaviour, you might choose to add to the list of rules (presumably with higher q-values than the default rules) and/or simply re-write them.

Below is a short description of the routing rules that are provided by default with the Gateway. For complete details, please refer to the rules' syntax in the *routing-rules.xml* file located at *[GATEWAY_HOME]\config\routing-rules.xml* (where *[GATEWAY_HOME]* is the root folder of the installation).

- **Rule "default_sip_out":** All incoming PSTN calls are sent to the SIP URI provided at installation time. The Caller ID, if present, is added in the 'From' URI using the format `sip:CallerID@GW_HOST_IP:GW_SIP_PORT`.

  This rule can be changed to modify the routing of PSTN calls to SIP applications, and/or to modify how the Caller ID information is presented.

- **Rule "default_pstn_out":** All incoming SIP calls that have a Request URI that matches the format `sip:<tel-number>@<gateway-host>:<gateway-ua-port>` are directed to the first available telephony port, where

  - `<tel-number>` is a telephony number to be dialed on the outbound telephony line
  - `<gateway-host>` is the name or IP address of the Gateway host
  - `<gateway-ua-port>` is the port number used by the Gateway's SIP user agent).

  This rule can be changed to modify the format or the field by which a SIP-based application directs outbound PSTN calls.

- **Rule "Redirect_to_sip":** A rule that is used to generate the outbound SIP call caused by a redirect instruction (message 3XX from a SIP server) to a SIP endpoint.
- **Rule "Redirect_to_pstn":** A rule used to generate a PSTN call (bridged on Gateway) when receiving a redirect instruction (message 3XX from a SIP server) to a PSTN phone number.

The role of the default routing rules is to provide basic functionality out of the box, as well as a sample of what can be achieved by customizing the data file.

TIP: Out of the box, the Gateway will route all incoming PSTN calls to the target SIP URI specified at installation time . If there is a SIP proxy in the the network, this is where the calls should be routed. To modify the default SIP destination, simply edit the routing rules file, find the rule named "*default_sip_out*", and modify the value of the outbound property *sip.out.requestUri*. To implement DNIS-based routing, please refer to Example 3: DNIS-based routing on page 175.

Caching of routing rules

Under high traffic conditions, the routing rules are not fetched on every incoming call. Instead, a caching mechanism is used to keep a temporary copy of the rules inside the Gateway. This cached version of the routing rules is kept for 10 seconds. After 10 seconds, a new *HTTP (HyperText Transfer Protocol)* query is performed. This mechanism helps limit the HTTP traffic.

The caching mechanism is automatically activated under high traffic conditions. Under low traffic conditions (that is, when handling a low number of calls per second), caching is automatically disabled.

The cache timeout value can be changed via the parameter *Netborder.gw.routing.cacheFlushTimeoutSec* in the global configuration file. When set to '0', the cache is disabled and the rules are fetched on every incoming call.

Note that if the fetched routing rules file is invalid or incorrectly formatted, the Gateway will generate a warning and use the previous valid routing rules for new incoming calls.

Chapter 8: Engineering guidelines

This chapter provides some essential guidelines or considerations.

This chapter contains the following topics:

IP network considerations

For all SIP legs established between the Gateway and an external SIP application, a full-duplex G.711 RTP session (2 x 64 Kbits/sec plus ~20% overhead for RTP packetization) will be established for the duration of the SIP session.

The Gateway is engineered to run on the same managed Local Area Network, Metropolitan Area Networks, or carefully engineered Wide Area Network as its target SIP applications. Deployments across the public Internet are not supported.

Telephony considerations

For optimal use of the resources, it is recommended that all telephony ports be configured as bi-directional. However, under such circumstances, a *glare* may arise.

Glare

A “glare” occurs on bi-directional ports when a port has been identified to carry out an outbound call (SIP-initiated call or bridge transfer request from the SIP application), and, before the call can actually take place, an inbound PSTN call is received. In these rare situations, the Gateway will give priority to the incoming call (accept the call), and then issue a 4XX response to the SIP application’s INVITE request.

To minimize occurrences of this scenario, select a “reverse linear” hunting algorithm for the telephony resources (starting with the last channel and going in reverse order of channel number), and ensure that the switch or PBX performs hunting on incoming PSTN calls starting with the first channel and going forward with increasing channel numbers. In this case, a glare condition could only occur when there is contention for the last available port in the system.

If a glare condition, however rare it may be, is totally unacceptable, then it is recommended that you dedicate ports for inbound (PSTN-originated) and others for outbound (SIP-initiated) calls. In addition, carefully engineer the system to ensure enough ports of both types are available.

Chapter 9: Troubleshooting

This chapter addresses key troubleshooting issues.

This chapter provides assistance with the following common problems:

- Unable to receive a PSTN call on page 185
- Unable to place a PSTN Call on page 185
- SIP application not answering calls from the Gateway on page 186.

Unable to receive a PSTN call

1. Make sure that the physical connectivity to the T1/E1 cable is correct.
2. Using the Web-based monitoring tool, check the status of the channels. See PSTN line status on page 92.
3. Verify warnings/errors in the Windows Event Viewer.

Unable to place a PSTN Call

1. Make sure that the physical connectivity to the T1/E1 cable is correct.
2. Using the Web-based monitoring tool, check the status of the channels. See PSTN line status on page 92.
3. Verify warnings/errors in the Windows Event Viewer.
4. Check the *[GATEWAY_HOME]/logs/GatewayDebug.out* top log file (see below).

When a call fails, the Gateway generates a message similar to the following:

```
INFO - Netborder.pstn.sangoma.telesoft.BoardChannel.s0c19.SM :
MakeCall operation failed. Cause=NO_MATCHING_RULE_CONN_FAILURE
```

In the example above, the string "`s0c19`" is specific to the channel where the call was attempted, and the "`NO_MATCHING_RULE_CONN_FAILURE`" string indicates the reason why the connection failed.

SIP application not answering calls from the Gateway

Ensure the audio formats and RTP packet size between the Gateway (*Netborder.rtp.encodingList* and *Netborder.media.rtp.packetSizeMs* parameters in the *gw.properties* file) and the target SIP application are compatible.

For example, the Gateway might offer A-law encoded audio to the target application set to accept only u-law. If that is the case, the SIP application will reject all calls with a SIP message '488 Not Acceptable Here'.

You can either change these platforms to use 20 ms (the Gateway's default) or change the *Netborder.media.rtp.packetSizeMs* parameter in the *gw.properties* file to '30' (assuming this value is used by the target application). Then restart the service.

Appendix A: Glossary

This appendix contains a list of abbreviations and acronyms used in this guide.

A

AAA	Authentication, Authorization, and Accounting
ACD	Automatic Call Distributor
AIS	Alarm Indication Signal
ANI	Automatic Number Identification
ASR	Automatic Speech Recognition

B

B2BUA	Back to Back User Agent
BRI	Basic Rate Interface (BRI)

C

CAS	Channel Associated Scripting
CCITT	International Telephone and Telegraph Consultative Committee (from the French name "*Comité consultatif international téléphonique et télégraphique*")

CDR	Call Detail Record
CNG	Comfort Noise Generation
CPA	Call Progress Analysis
CSP	Continuous Speech Processing
CSS	Composite Source Signal

D

DHCP	Dynamic Host Configuration Protocol
DNIS	Dialed Number Information Service
DSP	Digital Signal Processor
DTMF	Dual-Tone Multi-Frequency

E

eVAD	enhanced Voice Activity Detection

F

FTP	File Transfer Protocol

H

HTTP	HyperText Transfer Protocol

I

IE	Information Elements
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ITU	International Telecommunication Union
ITU-T	ITU Telecommunication Standardization Sector
IVR	Interactive Voice Response

L

LAN	Local Area Network

M

MAC	Media Access Control
ms	Milliseconds

N

NFAS	Non-Facility Associated Signaling

NLP	Non-Linear Processing
NT	Network Termination

O

OAM	Operations, Administration, and Maintenance
OAM&P	Operations, Administration, Maintenance and Provisioning
OEM	Original Equipment Manufacturer
OS	Operating System

P

PBX	Private Branch eXchange
PCM	Pulse Code Modulation
PCMA	PCM A-law
PCMU	PCM ulaw
PHB	Per-Hop-Behavior
PI	Progress Indicator
PRI	Primary Rate Interface
PSTN	Public Switched Telephone Network

R

RAI	Remote Alarm Indication
RFC	Request for Comments
RLT	Release Line Trunking
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol

S

SALT	Speech Application Language Tags
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SSRC	Synchronization Source

T

TBCT	Two B Channel Transfer

TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TE	Terminal Equipment
ToS	Type of Service

U

UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Data Protocol
UM	Unified Messaging
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

V

VAD	Voice Activity Detection
VoiceXML	Voice eXtensible Markup Language
VoIP	Voice over IP
VSM	Vital Sign Monitor

W

WSDL	Web Services Description Language

X

XML	eXtensible Markup Language

Appendix B: Configuration parameters

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

- Global configuration file (*gw.properties* file) on page 193
- PSTN configuration file (*pstn-config.xml*) on page 199.

For a detailed description of the routing rules file (*routing-rules.xml*), see Routing rule constructs on page 172.

For information on the SIP registration file, see SIP register configuration on page 139.

Global configuration file

The following table lists the configuration parameters contained in the global configuration file, *gw.properties,* file located at:

- *[GATEWAY_HOME]\config\gw.properties*

  where *[GATEWAY_HOME]* is the root folder of the installation (for example, *C:\Program Files\Netborder\Express\Gateway\\config\gw.properties)*

The *gw.properties* file is a simple text file that you can edit using any standard text editor. Alternatively, you can use the shortcut menu to view and edit the file directly from the *Netborder Express Gateway* entry accessible from your Programs list (**Start > Programs > Netborder Express Gateway > Configuration > Edit Global configuration file**).

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

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

Depending on the configuration selected during the installation process (e.g., analog, digital *CAS [Channel Associated Signaling]*, digital ISDN), different parameters may be available in the configuration file. Therefore, you will not see in your properties file all of the parameters listed here.

The default values should be suitable for most applications.

Global Configuration Properties (gw.properties)		
Parameter	**Default Value**	**Notes**
Netborder.gw.allLegConnectio n TimeoutMs	None	The maximum time that a call has to get connected. This includes ALL the legs that are tried in a call (see *Netborder.gw.maxRoutingRulesMatches*). Can **NOT** be overridden in the routing rules.
Netborder.gw.connection TimeoutMs	None	Timeout to get an 'answer', in milliseconds, after which time attempt to establish an outbound call leg is abandoned. This timeout starts after the 'ring' event has been received on the outbound leg. Triggers if no connection event ('200 OK' in SIP or 'Connected' signal in PSTN) has been received between the 'ring' event and the timeframe specified by this parameter. This parameter can be overridden in the routing rules.
Netborder.gw.maxRoutingRul es Matches	1	The maximum number of matching routing rules that are tried, in order of their q-value priority, to successfully establish a call. By default, only one routing rule that triggers, the one with the highest priority, is used to attempt a call.
Netborder.gw.ringTimeoutMs	None	Timeout to get a 'ring' (or an 'answer' if no ring is provided by remote end), in milliseconds, after which time attempt to establish an outbound call leg is abandoned. Triggers if no ring event (provisional response in VoIP or 'alerting' signal in PSTN) has been received within the specified timeframe. This parameter can be overridden in the routing rules.
Netborder.gw.routing.cache FlushTimeoutSec	10	The amount of time in seconds for determining if a new routing rule HTTP query should be performed. To disable routing rules caching, set this value to 0. Note that routing rules

Global Configuration Properties (gw.properties)		
Parameter	**Default Value**	**Notes**
		caching is active only under high traffic conditions. In low traffic conditions, no caching is performed.
Netborder.gw.routingRulesUrl	http://[hostname]:7782/config/routing-rules.xml	Routing rule URL location.
Netborder.media.ip.tos	184	IP *ToS (Type of Service)* field for RTP packets. Default value (184) is set to DiffServ Expedited Forwarding *PHB (Per-Hop-Behavior)*. See RFC 2598 for more information.
Netborder.media.rtp.initialDelayMs	20	Initial playback delay, in milliseconds, from the RTP source before audio is relayed to the PSTN channel. This initial jitter buffer value does not include the telephony buffer on the telephony board. A large value will minimize chances of lost audio before the call is answered on the PSTN side but will increase the initial latency.
Netborder.media.rtp.maxDelay Ms	60	Maximal playout delay, in milliseconds. Upon reaching this value, some audio is lost in order to reset the playout delay to Netborder.*media.rtp.initial DelayMs*.
Netborder.media.rtp.packetSizeMs	20	The size, in milliseconds, of each RTP packet. Can be set to 20 ms or 30 ms. **Note:** When changing the value using the RTP provider, adjust the *Netborder.media.initialDelayMs* parameter so that it is a multiple of the RTP packet size.
Netborder.media.rtp.rfc2833 Supported	true	Indicates whether RFC2833 DTMF relay is supported or not. Set this parameter to true to enable RFC2833 support, and to false to disable RFC2833 support.
Netborder.net.primaryIPAddress	None	The IP address for the local host to be used by the Gateway. By default, the Gateway resolves the localhost IP address programmatically; however, this parameter can be used to force a specific (valid) IP address for the host. Most useful for multi-home servers.

Global Configuration Properties (gw.properties)		
Parameter	**Default Value**	**Notes**
		The value of this parameter is used by the routing engine when routing rule variables such as *GW_HOST_IP* are expanded.
Netborder.oam.webServiceIp Address	None	IP address of the interface used by the OAM agent on the Gateway host. If not set, will bind to all local interfaces.
Netborder.oam.webServicePor t	18086	The port used by the OAM agent on the Gateway host.
Netborder.pstn.configFile	[GATEWAY_HOME] /config/pstn-config.xml	PSTN configuration file location.
Netborder.pstn.initialChannel State	in-service	Defines the state of all channels when the Gateway service starts. Valid values are: 'in-service' (default) and 'out-of-service'.
Netborder.rtp.encodingList	ulaw	A string of elements (comma separated) that represents the ordered list of media codecs supported by the Gateway, as per RFC 3551. The order determines the "preference" of the Gateway for a certain codec type, with the most preferred codec appearing first in the list. Elements of the list can be ulaw, alaw, g729, g726, g723.
Netborder.run.mode	development	Values : development|production. In **development** mode, call logging is enabled and caching schemes (HTTP and DNS clients) are disabled. Default logging level is INFO. Ideal mode for troubleshooting when call volume is low. In **production** mode, caching schemes are enabled for maximum performance. Call logging is disabled and the default logging level is WARN. Ideal mode for deployed applications.
Netborder.sip.clientRegistratio n. configFile	[GATEWAY_HOME] /config/sip-client-registration.xml	The location of the XML configuration file for the SIP registration parameters. This file is used to configure how the Gateway registers to a SIP registrar.
Netborder.sip.getHostByName TimeoutMs	250 ms	The maximum time that the Gateway will wait to resolve a host name's IP address. The default setting (250 ms) is convenient in the lab/development

Global Configuration Properties (gw.properties)		
Parameter	**Default Value**	**Notes**
		environment. **Note:** For production systems, depending on network characteristics, this value may need to be larger.
Netborder.sip.in.doNotSend180After183	false	For compatibility with some SIP user agents. If true, a 180 response with SDP will not be sent after a 183 response.
Netborder.sip.in.override1XXWithSDP		For compatibility with some SIP user agents. If set, a 180 response with SDP will not be sent; however, the code provided (usually 183) will be used. Could be used in the reverse logic for overriding a 183 response with SDP with a 180 response.
Netborder.sip.out.acceptEarlyMedia	true	Defines if the Gateway will start the media prior call connection upon reception of a 1XX message with SDP (Session Description Protocol).
Netborder.sip.localOutboundProxyURI	None	The URI to use for all outgoing SIP INVITEs. Note that when an outbound proxy is used, the Request URI in the outgoing SIP INVITE is a “logical” URI destined for the end application. The physical address where the message is sent is that of the outbound proxy.
Netborder.sip.userAgent.IPAddress	INADDR_ANY:5066/udp INADDR_ANY:5066/tcp	A comma-separated list of IP addresses and ports to which the application should listen for incoming SIP calls. Format : ip_addr1[:port1][/[udp|tcp]], ip_addr2[:port2][/[udp|tcp]] Use *INADDR_ANY* as the IP address to listen on all IP interfaces on the host.
Netborder.sip.includeRportParameterInViaHeader	true	When true, SIP request will contain the rport parameter in the via header. For more details about rport usage, please refer to IETF RFC-3581 - An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing.
Netborder.win32ProcessPriority	REALTIME	The priority of the Gateway service in the Windows OS. By default, this is set to ‘REALTIME’. The priority can be

Global Configuration Properties (gw.properties)		
Parameter	**Default Value**	**Notes**
		increased to ‘HIGH’ or ‘REALTIME’.

PSTN configuration file

This section lists and describes the configuration parameters contained in the PSTN configuration file, *pstn-config.xml,* located at:

- *[GATEWAY_HOME]\config\pstn-config.xml*

  where *[GATEWAY_HOME]* is the root folder of the installation (for example, C:\Program Files\Netborder\Express\Gateway*config\pstn-config.xml*

The Relax NG XML schema can be found at:

- *[GATEWAY_HOME]\config\pstn-config.rng*

XML structure

PSTN configuration is described using an XML format. The data structure of the XML file is outlined below. Note that in this example, elements have been stripped of attributes to provide an overview of the structure only; specifically, to give you an idea of parent and child relationships. For a complete sample PSTN configuration file, refer to Appendix E.

For a description of the elements contained in the PSTN configuration file, as well their attributes, see the PSTN Configuration Parameters Table.

```
<?xml version="1.0" encoding="utf-8"?>
<pstnConfig version="1.0">

 <!-- List of all PSTN interface providers -->
 <ifProviders>
   <sangoma>
   </sangoma>
 </ifProviders>
 <!-- End of list of all PSTN interface providers -->

 <!-- All PSTN interfaces -->
 <interfaces>
   <sangoma>
     <BChannels />
     <Dchannel />
   </sangoma>
```

```
  <sangoma>
    <BChannels>
      <VoiceQualityEnhancement>
        <EchoCancellation />
        <AcousticEchoCancellation />
        <LevelControl />
        <BackgroundNoiseAttenuation />
        <DTMFRemoval />
      </VoiceQualityEnhancement>
    </BChannels>
    <Dchannel />
  </sangoma>
</interfaces>
<!-- End of PSTN interfaces -->

<!-- Call control parameters -->
<callControl>

  <!--Resources groups -->
  <resourcesGroups>
    <resourcesGroup />
    <resourcesGroup />
  </resourcesGroups>
  <!-- End of Resources groups -->

  <!-- ISDN parameters -->
  <isdn>

    <!-- Single interfaces ISDN groups -->
    <groups>
      <group>
        <member />
      </group>
      <group>
        <member>
          <BChannels>
            <BChannelRange>
              <resourcesGroup />
              <resourcesGroup />
            </BChannelRange>
            <BChannelRange>
              <resourcesGroup />
            </BChannelRange>
          </BChannels>
        </member>
      </group>
```

```
    </groups>
    <!-- End of Single interface groups -->

    <!-- NFAS ISDN groups -->
    <nfasGroups>
      <nfasGroup>
        <members>
          <member>
            <BChannels>
              <BChannelRange>
                <resourcesGroup />
              </BChannelRange>
            </BChannels>
          </member>
          <member>
            <BChannels>
              <BChannelRange>
                <resourcesGroup />
              </BChannelRange>
            </BChannels>
          </member>
        </members>
      </nfasgroup>
    </nfasGroups>
    <!-- End of NFAS groups -->

  </isdn>
  <!-- End of ISDN -->

 </callControl>
 <!-- End of call Control -->

</pstnConfig>
```

This table lists PSTN configuration elements and provides a brief description as well as the relevant attributes, where applicable.

PSTN Configuration Top Level Parameters		
Element	***Description***	***Attributes***
<pstnConfig>	Parent element for the PSTN configuration file.	• **version** (decimal): Version number of the PSTN configuration.
<ifProviders>	Contains a list of all PSTN interface providers.	
<sangoma>	Contains general parameters specific to Sangoma.	For details refer to table Sangoma ifProvider Parameters on page 203.
<interfaces>	Contains a list of all PSTN interfaces.	
<sangoma>	Contains the paramters for one sangoma interface.	For details refer to table Sangoma interface Parameters on page 204.
<callControl>	Parent element for call control parameters	
<resourcesGroups>	Contains individual resources groups. Resources groups are used for inbound and outbound B channel reservation.*	For details refer to table Resources Groups Parameter on page 210.
<isdn>	Contains the ISDN parameters for each ISDN interface (group).	For details refer to table Resources Groups Parameter on page 210.

Sangoma ifProvider Parameters		
Element	***Description***	***Attributes***
<BChannelsThreads>	Number and priority of threads used to process B channels.	• **count** (positive integer): Number of threads. Recommended value is two times the number of CPUs. • **priority**: Priority of the threads. Values include LOWEST, BELOW_NORMAL, NORMAL, ABOVE_NOMAL, HIGHEST, TIME_CRITICAL. **Recommended value: TIME_CRITICAL.**
<DChannelsThreads>	Number and priority of threads used to process D-channel data.	• **count** (positive integer): Number of threads. Recommended value is '1'. • **priority**: Priority of the threads. Valid values include LOWEST, BELOW_NORMAL, NORMAL, ABOVE_NOMAL, HIGHEST, TIME_CRITICAL. **Recommended value: HIGHEST.**

Sangoma interface Parameters		
Element	***Description***	***Attributes***
<sangoma>	One such element is required per interface. For example, if you installed four Sangoma interfaces), you will need four <sangoma> elements.	• **ID** (string): Unique identifier for the interface. • **wanpipe** (positive integer): Number given to that interface by the Sangoma device driver. • **mediaType**: T1 (for North America, Japan) or E1 (the rest of the world). • **framing**: T1/E1 framing types. Valid values for T1 are ESF, ESF_Japan, D4. Valid values for E1 are CRC4, NonCRC4. • **lineEncoding**: T1/E1 line encoding type. Valid values for T1 are AMI, B8ZS. Valid values for E1 are AMI, HDB3. • **LBO** (T1 only): Line Built-in Output. Applies only when 'mediaType' is T1. For valid values, see Set T1 cable length on page 112. • **clocking:** Valid values are TERMINAL (to act as terminal equipment and receive the clock reference from the network) and NETWORK (to act as network equipment like a PSTN switch). • **E1Signaling** (E1 only): E1 signaling type. Applies only when 'mediaType' is E1. Set to CSS when the interface is used by an ISDN-PRI signaling group and set to CAS when the interface is used by CAS signaling group. Valid values are NONE, CSS, CAS. • **E1Termination** (E1 only): Applies only when 'mediaType' is E1. Valid values are 120ohm and 75ohm. • **loopbackMode** (optional): Loopback mode for the interface. Current value is

Sangoma interface Parameters		
Element	***Description***	***Attributes***
		DISABLED.
<BChannels>	Element that contains attributes that both control the media format and activate the echo cancellation feature on the PSTN interface.	• **defaultPcmLaw**: Default PCM law for all B channels of the interface. Valid values are PCMA (PCM A-law) and PCMU (PCM ulaw). • **voicePacketLengthInMs** (positive integer between 1 and 31): Length in milliseconds of the voice packets received or transmitted from/to this interface. • **echoCancellationMode**: Indicates if echo cancellation is enabled for all B channels of the interface. Valid values are DISABLED and NORMAL.
<VoiceQualityEnhancement>	Parent element that control the element that control various features that improve the voice quality.	
<EchoCancellation>	Element that contains the attributes that controls the behavior of the echo cancellation for on all B-Channels if the current interface. Simply put, echo is the phenomenon of hearing one's own voice, with a delay. The larger this delay, the more discomfort experienced by the individual hearing it. Beyond a certain threshold, conversation is	• **mode** select the operational mode of the echo canceller. Valid values are: NORMAL for most situations where echo cancellation is required; SPEECH_RECOGNIZER_FRIENDLY for systems where speech recognizer engines are used (in this mode, the Non Linear Processor - NLP is disabled); DISABLED for systems that does not need echo cancellation. • **comfortNoiseMode** valid values are: NORMAL to inject comfort noise to the traffic sent to the the RTP stream, DISABLED to disable comfort noise injection. • **tailDisplacementInMs**

Sangoma interface Parameters		
Element	**Description**	**Attributes**
	impeded and becomes unnatural. Echo Cancellation devices provide a means to remove echo, as it is detrimental to communication. The ITU defines echo cancellation as follows: “Echo cancellers are voice operated devices placed in the 4-wire portion of a circuit (which may be an individual circuit path or a path carrying a multiplexed signal) and are used for reducing the echo by subtracting an estimated echo from the circuit echo.” (ITU-T G.168)	(unsigned integer between 0 and 896): The echo tail-displacement allows the user to compensate for the fixed network delays and been able to cancel echo for a window larger than 128ms. • **doubleTalkBehavior** This parameters allows to control how the echo canceller behaves when both sides of call are talking simultaneously. Valid values are: OPTIMAL use this value for most application, MORE_RESIDUAL_ECHO use this value if your application does not perform well because the echo canceller clips the voice to often in your application when btoh ends of the call are talking simultaneously (could be required in application that supports barge-in), MORE_CLIPPING use this value if your application not need double talk support and improve the echo cancellation.
<AcousticEchoCancellation>	Element that contains the attributes that controls the behaviour of the acoustic echo canceller. Acoustic echo is generated through a telephone set when sounds directly picked up by the microphone also travel within the location of the microphone, reflect on objects at that	• **mode**. Valid values are NORMAL to enable the acoustic cancellation and DISABLED to disable the acoustic echo cancellation.

Sangoma interface Parameters		
Element	***Description***	***Attributes***
	location, and are then indirectly picked up again, with a certain amount of delay, by the microphone. The sounds may initiate from either party as well as from any other ambient noise in the location, including the telephone set speaker (ear piece).	
<LevelControl>	Element that contains the attributes to control the level of the voice signal sent to the RTP or to the PSTN.	• **mode**. The valid values are: MANUAL to increase or decrease the voice level by a constant gain (see); AUTOMATIC to get voice level to specified level (see); DISABLED to disabled any voice level control. • **direction** valid values are TO_PSTN to enable level control on the voice signal sent to PSTN; FROM_PSTN to enable level control on the voice signal received from the PSTN before sending it to RTP; ALL to enable level control to both directions. • **gainIndB** (integer value between 40 and -40). This value is used when the _ attribute is set to MANUAL. Set to a positive value to increase the voice level. Set to a negative value to decrease the voice level. Set to 0 to keep the voice level unchanged. • **targetLevelIndBm0** (integer value between 0 and -40). Set to 0 to amplified the signal to its maximum value and -40 attenuate the signal to a minimum value. The recommended value is -18 dBM0. This parameter is used

Sangoma interface Parameters		
Element	**Description**	**Attributes**
		when the attribute is set to AUTOMATIC. The ALC algorithm is designed to adjust the level of the talker to a user-defined target energy level. The gain applied can be negative or positive, resulting in either amplification or attenuation of the signal, depending on the target level. While the gain is applied on the whole signal (i.e. voice and background noise), only the energy of the talker is used when deciding to amplify or reduce the signal. For this purpose, a voice activity detector specific to the ALC algorithm (ALC VAD) is used to differentiate between voice and noise signals.
<BackgroundNoiseAttenuation>	Element that contains the attributes to control the background noise attenuation feature.	• **mode** valid values NORMAL to enable background noise attenuation on the voice signal; DISABLED to disable this feature. • **direction** valid value is FROM_PSTN to enable background noise attenuation on the voice signal sent received from the PSTN before sending it to RTP.
<DTMFRemoval>	Element that contains the attributes to control the removal if the DTMF signal sent to the RTP leg.	• **mode** valid values are: ENABLED to remove DTMF signal from the voice stream when a DTMF signal is detected; DISABLED to disable this feature.
<DChannel>	Optional element that contains attributes that describe both the position and maximum transmission unit of the D channel in the interface. Optional.	• **channelIndex** (positive integer between 1 and 31): Position of the D channel in the interface. Set to 0 to disable the D channel. Valid T1 D-channel positions are [1-24], and the usual D-Channel position is 24. Valid E1 D-channel positions are [1-31], and the usual D-Channel

Sangoma interface Parameters		
Element	**Description**	**Attributes**
		position is 16. • **mtuSize** (positive integer between 256 and 4096): D-Channel packet Maximum Transmission Unit (MTU). Refers to the size (in bytes) of the largest packet that a Q921 layer of a communications protocol can pass onward. **Recommended value is 2048.**

Resources Groups Parameter		
Element	***Description***	***Attributes***
<resourcesGroup>	One such element is required per resources group.	• **ID** (string): Unique identifier for the resources group. • **direction**: Indicates the direction for which this group of resources will be used: IN for incoming calls, OUT for out going calls, and BIDIR for both incoming and outgoing calls. • **outboundHuntingScheme**: Indicates how the channel hunting will be performed on this group for outbound calls. Must be present if direction is OUT or BIDIR. Valid values are LINEAR and REVERSE LINEAR. • **inboundHuntingScheme**: Indicates how channel hunting will be performed on this group for inbound calls. Must be present if direction is IN or BIDIR. Valid values are LINEAR and REVERSE LINEAR.

ISDN Parameters		
Element	***Description***	***Attributes***
<groups>	List of all ISDN groups containing a single PSTN interface where each interface has a dedicated D-Channel carrying ISDN signalling. This element could be omitted if your application does not require such groups.	For details refer to table ISDN Group Parameters on page 212.
<nfasGroups>	List of all ISDN groups containing a multiple PSTN interfaces where one interface is designed to have a D-Channel carrying the ISDN signalling for all interface within the group. This element could be omitted if your application does not require such groups.	For details refer to table NFAS Group Parameters on page 216.

ISDN Group Parameters		
Element	***Description***	***Attributes***
<group>	A group can contain one interface only. One such element is required per group.	• **ID** (string): Unique identifier for the ISDN group. • **termination**: Indicates if this group must act as a NETWORK or as a TERMINAL endpoint. Set this value to TERMINAL if this group connects to a PSTN provider. This is the most common value. • **switchVariant**: Name of the switch variant to use with this group. Valid values are: EuroISDN, DMS100, 5ESS, NI2. • **BChannelNegotiation**: Indicates how the B Channels are negotiated. Valid values are: PREFERRED and EXCLUSIVE. • **initiateRestartProcedure:** Indicates if the gateway shall initiate or not the ISDN restart procedure on the current group when the layer 2 is coming up. Valid values are: YES and NO. This parameter is optional its default value is: YES. • **inBandProgressTonesGeneration:** Indicates when in-band progress tones should be generated on inbound ISDN calls. Valid values are: AS-NEEDED, NEVER and ALWAYS.When this parameter is set to AS-NEEDED, the gateway will generate in-band progress tones only when request by the remote end in the ISDN setup message. When this parameter is set to ALWAYS the gateway will generate in-band progress on every call coming on the current ISDN group. When set to NEVER, the gateway will never generate in-band

ISDN Group Parameters		
Element	***Description***	***Attributes***
		progress tone. This parameter is optional its default value is: AS-NEEDED. • **inBandProgressTonesIndicator:** This parameter indicates which ISDN message(s) shall be used to notify the remote end when the gateway generate in-band progress tones. Valid values are: ALERTING, PROGRESS and BOTH. When this parameter is set to ALERTING, the gateway will send only the ALERTING message. When this parameter is set to PROGRESS the gateway will only send the PROGRESS message. When this parameter is set to BOTH the gateway will send the PROGRESS message followed by the ALERTING message. Of course the message are containing the progress indicator information element. This parameter is optional its default value is: PROGRESS.
<member>	PSTN interface associated with this group.	• **interfaceID** (string): ID of the PSTN interface associated with this group. This ID must match the ID of an interface listed above under <PSTNConfig><interfaces>.
<BChannels>	Contains a list of active B Channels of the group.	
<BChannelRange>	Contains a range of B Channels. Valid T1 B-channel positions are [1-24]. Valid E1 B-channel positions are [1-30]. A position shall appear in only one range and the D-channel position is	• **firstchannel** (positive integer): Values range from 1 to 30. • **lastChannel** (positive integer): Values range from 1 to 30.

ISDN Group Parameters		
Element	**Description**	**Attributes**
	not allowed.	
<resourcesGroup>	Name of the resources group to which this channel belongs. Please refer to the ID attribute of the resourcesGroup element: <callControl><resourcesGroups><resourcesGroup>	string

NFAS Group Parameters		
Element	***Description***	***Attributes***
<nfasgroup>	A NFAS group can contain multiple interfaces. One such element is required per NFAS group.	• **ID** (string): Unique identifier for the ISDN group. • **termination**: Indicates if this group must act as a NETWORK or as a TERMINAL endpoint. Set this value to TERMINAL if this group connects to a PSTN provider. This is the most common value. • **switchVariant**: Name of the switch variant to use with this group. Valid values are: EuroISDN, DMS100, 5ESS, NI2. • **BChannelNegotiation**: Indicates how the B Channels are negotiated. Valid values are: PREFERRED and EXCLUSIVE. • **initiateRestartProcedure:** Indicates if the gateway shall initiate or not the ISDN restart procedure on the current group when the layer 2 is coming up. Valid values are: YES and NO. This parameter is optional its default value is: YES. • **inBandProgressTonesGeneration:** Indicates when in-band progress tones should be generated on inbound ISDN calls. Valid values are: AS-NEEDED, NEVER and ALWAYS.When this parameter is set to AS-NEEDED, the gateway will generate in-band progress tones only when request by the remote end in the ISDN setup message. When this parameter is set to ALWAYS the gateway will generate in-band progress on every call coming on the current ISDN group. When set to NEVER, the gateway will never generate in-band progress tone. This parameter is

NFAS Group Parameters		
Element	***Description***	***Attributes***
		optional its default value is: AS-NEEDED. • **inBandProgressTonesIndicator:** This parameter indicates which ISDN message(s) shall be used to notify the remote end when the gateway generate in-band progress tones. Valid values are: ALERTING, PROGRESS and BOTH. When this parameter is set to ALERTING, the gateway will send only the ALERTING message. When this parameter is set to PROGRESS the gateway will only send the PROGRESS message. When this parameter is set to BOTH the gateway will send the PROGRESS message followed by the ALERTING message. Of course the message are containing the progress indicator information element. This parameter is optional its default value is: PROGRESS.
<members>	List of all member interfaces forming this NFAS group.	
<member>	A PSTN interface associated with this group.	• **interfaceID** (string): ID of the PSTN interface associated with this group. This ID must match the ID of an interface listed above under <PSTNConfig><interfaces>. • **nfasInterfaceID** (unsigned number): Interface ID used the ISDN protocol to identify this interface within the NFAS group. This value shall be unique among all member of a given NFAS group. • **DChannelMode**: Indicates the whether this interface carries ISDN signalling or not. Valid

NFAS Group Parameters		
Element	***Description***	***Attributes***
		values are: PRIMARY and DISABLED. Within a given NFAS group only one member shall specify PRIMARY. All other members channel specify DISABLED.
<BChannels>	Contains a list of active B Channels of the group.	
<BChannelRange>	Contains a range of B Channels. Valid T1 B-channel positions are [1-24]. Valid E1 B-channel positions are [1-30]. A position shall appear in only one range and the D-channel position is not allowed.	• **firstchannel** (positive integer): Values range from 1 to 30. • **lastChannel** (positive integer): Values range from 1 to 30.
<resourcesGroup>	Name of the resources group to which this channel belongs. Please refer to the ID attribute of the resoucesGroup element: <callControl><resourcesGroups><resourcesGroup>	string

*To add a range of B channels in a given resources group, specify the range when the B channel range is specified in an ISDN group.
A given B channel can appear in multiple resources groups. However, as soon as that B channel is used it becomes unavailable to other resources groups until it is released.

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:

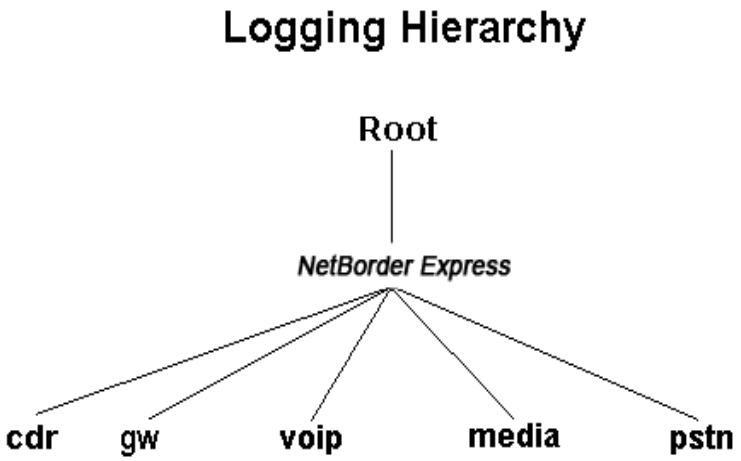
- **FATAL:** Logs very severe error events that may lead the application to abort.
- **ERROR:** Logs only error conditions. The ERROR level provides the smallest amount of logging information.
- **WARN:** Logs information when an operation completes successfully but there are issues with the operation.
- **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 Gateway’s logging hierarchy. Those components that are used solely for troubleshooting purposes by Technical Support have not been included (for example, all of the Gateway’s infrastructure components and utilities).



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

- **PSTN:** Relates to everything that has to do exclusively with telephony signaling within the context of a PSTN call leg.
- **Media:** Relates to everything that has to do exclusively with media processing (RTP packetization, audio format, echo cancellation, DTMF relay, etc.).
- **VoIP:** Relates to everything that has to do exclusively with signaling within the context of a VoIP call leg.

- **GW:** Relates to everything that has to do with the general Gateway application and is independent of the specifics of the PSTN, Media, VoIP implementations. It includes, for example, the Call Engine and Routing Engine logs.
- **CDR:** Contains high-level information about a call (such as call start time, call end time, ANI, DNIS, etc.), and can be used to generate billing information.

Configuring the logging subsystem

To configure the logging subsystem, follow these steps:

- Step 1: Set the logging level and appender (see page 224)
- Step 2: Set the pattern layout (see page 226)
- Step 3 (optional): Set child-specific behaviour (see page 227)

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. These parameters would be included in the file *[GATEWAY_HOME]\config\gw.properties* (where *[GATEWAY_HOME]* is the root folder of the installation).

```
# 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). Applies only when the Gateway is started from a Console.
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=Gateway.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=Gateway.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=Gateway.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. Again, these parameters are included in the file *[GATEWAY_HOME]\config\gw.properties* (where *[GATEWAY_HOME]* is the root folder of the installation).

```
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 event (such as a global configuration parameter or routing rule), 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).

Pattern Layout Conversion Characters	
Conversion Character	***Description***
%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 configured for the Gateway 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.

To enable the call logger, make the following changes to the *gw.properties* file:

```
# Logger configuration for the Windows Event log, level = INFO
# Also attach the CALL_LOG_APPENDER to duplicate call-specific
# information to a unique file
log4cplus.rootLogger=INFO, NTEVENTLOG, CALL_LOG_APPENDER
log4cplus.appender.CALL_LOG_APPENDER.Directory=test-output
log4cplus.appender.CALL_LOG_APPENDER.ImmediateFlush=false
log4cplus.appender.CALL_LOG_APPENDER.layout=log4cplus::PatternLayout
log4cplus.appender.CALL_LOG_APPENDER.layout.ConversionPattern=%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=false
```

If *log4cplus.appender.CALL_LOG_APPENDER.Directory* is set to a valid output directory (with write permission), call logs are written to that directory. Otherwise, call logging is disabled. If, on the other hand, *Netborder.infra.CallLogger.dateTimeDirectory* is set to “*true”*, a sub-directory structure with a year, month, day and hour hierarchy is 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 the 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.

Currently, the following loggers use the *CALL_LOG_APPENDER*:

- **Netborder.cdr.pstn:** Call start/end, ANI and DNIS at *INFO_LOG_LEVEL*. Can be used for generating simple billing information.
- **Netborder.pstn:** All logging statements that take place between call start and call end can be duplicated to the call log, provided that call logging has been enabled and that the logging level of the *Netborder.pstn* logger is high enough. Used for troubleshooting purposes.

To enable all instrumented call logs to appear in the separate file, make sure the *CALL_LOG_APPENDER* is attached to the root logger. For CDR only, ensure that the *Netborder.cdr* logger has this appender enabled at 'INFO' level or more. See the example below.

```
# Logger configuration for the CDR logger, level = INFO
# Attach the CALL_LOG_APPENDER to duplicate call-specific
# high-level information to a unique file

log4cplus.logger.Netborder.cdr=INFO, CALL_LOG_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, amend the *gw.properties* file 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. 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: Sangoma boards

This appendix contains a description of the Sangoma boards currently supported by the Gateway.

A101 (single T1/E1)

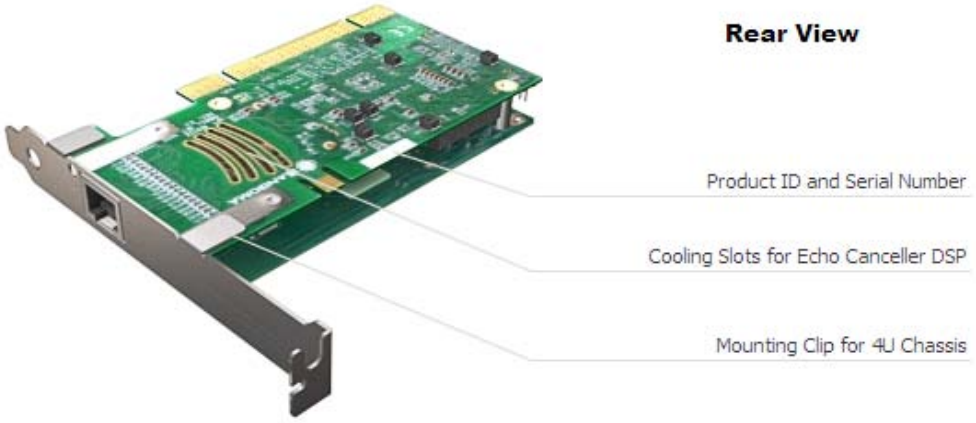
The Sangoma A101 board has one port of optimized voice and data over T1, E1, and/or J1. Like all boards in the Sangoma Advanced Flexible Telecommunications product line, the Sangoma A101 is equipped with “crash proof”, field upgradeable firmware.

The Sangoma A101 board:

- Supports up to 30 voice calls or 2.048 Mbps of full-duplex data throughput over two T1, E1, or J1 lines.
- Contains two T1/E1 ports with a single PCI or PCI-Express interface optimized for high performance voice and data applications.
- Is fully compatible with all commercially available motherboards—proper PCI-standard interrupt sharing without manual tuning.

For a complete description including technical specifications, read Sangoma’s product sheet at the following URL:

- http://www.sangoma.com/datasheets/p_a101-specs



A102 (dual T1/E1)

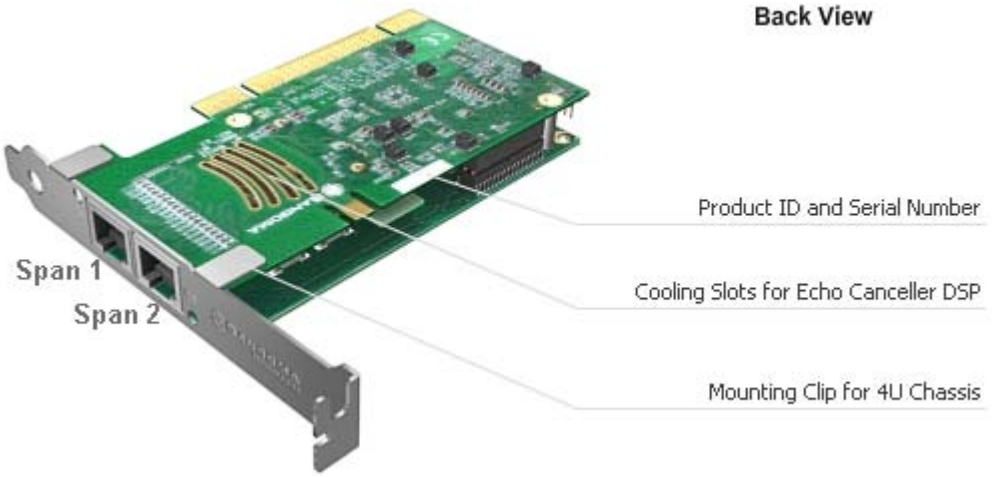
The Sangoma A102 board has two ports of optimized voice and data over T1, E1, and/or J1. Like all boards in the Sangoma Advanced Flexible Telecommunications product line, the Sangoma A102 is equipped with “crash proof”, field upgradeable firmware.

The Sangoma A102 board:

- Supports up to 60 voice calls or 4.096 Mbps of full-duplex data throughput over two T1, E1, or J1 lines.
- Contains two T1/E1 ports with a single PCI or PCI-Express interface optimized for high performance voice and data applications.
- Is fully compatible with all commercially available motherboards—proper PCI-standard interrupt sharing without manual tuning.

For a complete description including technical specifications, read Sangoma’s product sheet at the following URL:

- http://www.sangoma.com/datasheets/p_a102-specs



A104 (quad T1/E1)

The Sangoma A104 board has four ports of optimized voice and data over T1, E1, and/or J1. Like all boards in the Sangoma Advanced Flexible Telecommunications product line, the Sangoma A104 is equipped with “crash proof”, field upgradeable firmware.

The Sangoma A104 board:

- Supports up to 120 voice calls or 8.192 Mbps of full-duplex data throughput over four T1, E1, or J1 lines.
- Contains T1/E1 ports with a single PCI or PCI-Express interface optimized for high performance voice and data applications.
- Is fully compatible with all commercially available motherboards—proper PCI-standard interrupt sharing without manual tuning.

For a complete description including technical specifications, read Sangoma’s product sheet at the following URL:

- http://www.sangoma.com/datasheets/p_aft-104-specs

A108 (octal T1/E1)

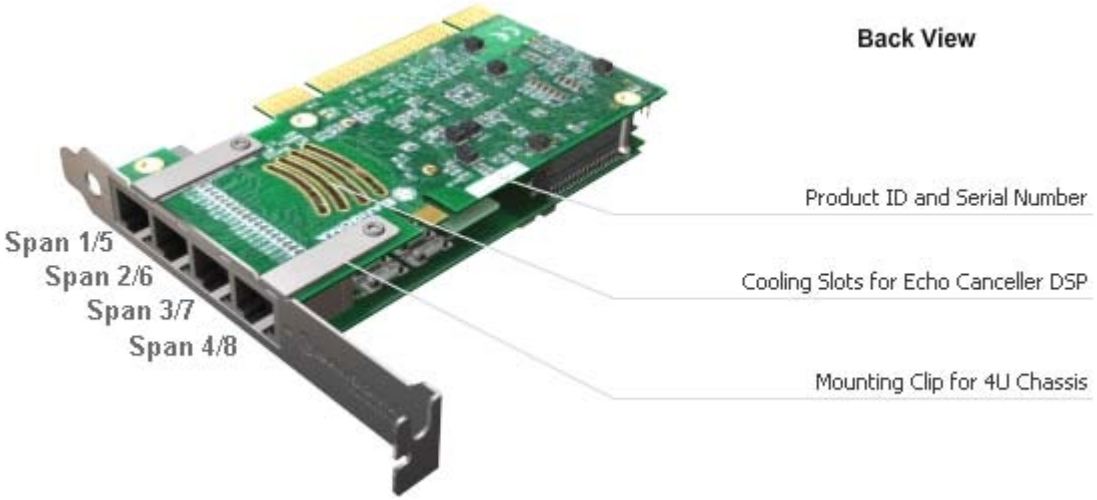
The Sangoma A108 Card has eight ports of optimized voice and data over T1, E1, and/or J1. Like all boards in the Sangoma Advanced Flexible Telecommunications product line, the Sangoma A108 is equipped with “crash proof”, field upgradeable firmware.

The Sangoma A108 board:

- Provides up to 16.4Mbps of full duplex data throughput or 240 voice calls over eight T1 and or E1 lines.
- Contains T1/E1 ports with a single PCI or PCI-Express interface optimized for high performance voice and data applications.
- Ships with “port-splitter” cables. Each RJ45 connection on the board supports two physical T1/E1 ports.
- Is fully compatible with all commercially available motherboards—proper PCI-standard interrupt sharing without manual tuning.

For a complete description including technical specifications, read Sangoma’s product sheet at the following URL:

- http://www.sangoma.com/datasheets/a108-specs



Appendix E: PSTN configuration file examples.

This appendix contains a some examples of PSTN configuration files.

- 1 T1 configured to be connected on a 5ESS switch on page 243
- 1 E1 configured to be connected on a EuroISDN switch on page 246
- 2 T1s configured to be connected on an NI2 switch in NFAS configuration on page 249.

For more detailed information on the PSTN configuration file, see the following topics located elsewhere in this guide:

- Chapter 5: PSTN
- PSTN configuration file on page 199.

Example 1: 1 T1 configured to be connected on a 5ESS switch

```
<?xml version="1.0" encoding="utf-8"?>
<!-- RelaxNG shema=pstn-config.rng -->
<pstnConfig version="1.0">

 <!-- List all PSTN interface providers -->
 <ifProviders>
  <sangoma>

   <!-- General parameters specific to Sangoma -->
   <BChannelsThreads count="4"
                priority="TIME_CRITICAL"/>
   <DChannelsThreads count="1"
                priority="HIGHEST"/>
  </sangoma>
 </ifProviders>

 <!-- All PSTN interfaces -->
 <interfaces>
  <sangoma id="XYZspan1"
      wanpipe="1"
      mediaType="T1"
      framing="ESF"
      lineEncoding="B8ZS"
      LBO="0dB"
      clocking="TERMINAL"
      loopbackMode="DISABLED">
     <BChannels defaultPcmLaw="PCMU"
            voicePacketLengthInMs="20">
     <VoiceQualityEnhancement>
      <EchoCancellation mode="NORMAL"
                  comfortNoiseMode="NORMAL"
                  tailDisplacementInMs="0"
                  doubleTalkBehavior="OPTIMAL"/>
      <AcousticEchoCancellation mode="NORMAL"/>
      <LevelControl direction="ALL"
                mode="AUTOMATIC"
                targetLevelIndBM0="-20"/>
      <BackgroundNoiseAttenuation direction="FROM_PSTN"
                        mode="ENABLED"
                        attenuationIndB="-18" />
```

```
        <DTMFRemoval direction="FROM_PSTN"
                     mode="DISABLED"/>
      </VoiceQualityEnhancement>
    </BChannels>
    <DChannel channelIndex="24"
              mtuSize="2048"/>
  </sangoma>
 </interfaces>

<!-- Call control parameters -->
 <callControl>
  <!-- Resources groups

   - Resources groups are used for inbound and outbound BChannel reservation.
   - Resources groups contains a list of BChannels.  To add a range of BChannel
     in a given resources group specify it when the BChannel range is specified
     in a ISDN group or ISDN NFAS group.
   - A given BChannel channel can appear in multiple resources group.  But as
     soon as that B-Channel is used it becomes unavailable in every resources
     groups until it is released.

  -->
  <resourcesGroups>

    <resourcesGroup ID="default"
                    direction="BIDIR"
                    outboundHuntingScheme="REVERSE_LINEAR"
                    inboundHuntingScheme="REVERSE_LINEAR"/>

  </resourcesGroups> <!-- End of Resources groups -->

  <!-- ISDN configuration

   - Contains the ISDN parameters for each ISDN interface (group) or group of
     interface (nfasGroup).
   -->
  <isdn>

    <!-- Call Control groups
      - A group can contain one interface.  When a group contains more
        than one interface this is a nfasGroup
     -->
    <groups>
     <group ID="XYZspan1"
            termination="TERMINAL"
            switchVariant="5ESS"
```

```
            BChannelNegotiation="EXCLUSIVE">
        <member interfaceID="XYZspan1">
          <BChannels>
            <BChannelRange firstChannel="1" lastChannel="23">
              <resourcesGroup>default</resourcesGroup>
            </BChannelRange>
          </BChannels>
        </member>
      </group>

    </groups> <!-- End of groups -->

  </isdn> <!-- End if ISDN -->

 </callControl> <!-- End of call Control -->

</pstnConfig>
```

Example 2: 1 E1 configured to be connected on an EuroISDN switch

```
<?xml version="1.0" encoding="utf-8"?>
<!-- RelaxNG shema=pstn-config.rng -->
<pstnConfig version="1.0">

 <!-- List all PSTN interface providers -->
 <ifProviders>
  <sangoma>

   <!-- General parameters specific to Sangoma -->
   <BChannelsThreads count="4"
                priority="TIME_CRITICAL"/>
   <DChannelsThreads count="1"
                priority="HIGHEST"/>
  </sangoma>
 </ifProviders>

 <!-- All PSTN interfaces -->
 <interfaces>
  <sangoma id="XYZspan1"
      wanpipe="1"
      mediaType="E1"
      framing="D4"
      lineEncoding="HDB3"
      clocking="TERMINAL"
      E1Signaling="CCS"
      E1Termination="120ohm"
      loopbackMode="DISABLED">
   <BChannels defaultPcmLaw="PCMA"
           voicePacketLengthInMs="10">
     <VoiceQualityEnhancement>
      <EchoCancellation mode="NORMAL"
                   comfortNoiseMode="NORMAL"
                   tailDisplacementInMs="0"
                   doubleTalkBehavior="OPTIMAL"/>
      <AcousticEchoCancellation mode="NORMAL"/>
      <LevelControl direction="ALL"
                mode="AUTOMATIC"
                targetLevelIndBM0="-20"/>
      <BackgroundNoiseAttenuation direction="FROM_PSTN"
                          mode="ENABLED"
```

```
                        attenuationIndB="-18" />
      <DTMFRemoval direction="FROM_PSTN"
                mode="DISABLED"/>
    </VoiceQualityEnhancement>
  </BChannels>
  <DChannel channelIndex="16"
          mtuSize="2048"/>
 </sangoma>
</interfaces>

<!-- Call control parameters -->
<callControl>
 <!-- Resources groups

  - Resources groups are used for inbound and outbound BChannel reservation.
  - Resources groups contains a list of BChannels.  To add a range of BChannel
    in a given resources group specify it when the BChannel range is specified
    in a ISDN group or ISDN NFAS group.
  - A given BChannel channel can appear in multiple resources group.  But as
    soon as that B-Channel is used it becomes unavailable in every resources
    groups until it is released.

 -->
 <resourcesGroups>

  <resourcesGroup ID="default"
              direction="BIDIR"
              outboundHuntingScheme="REVERSE_LINEAR"
              inboundHuntingScheme="REVERSE_LINEAR"/>

 </resourcesGroups>
 <!-- End of Resources groups -->

 <!-- ISDN configuration

  - Contains the ISDN parameters for each ISDN interface (group) or group of
    interface (nfasGroup).
  -->
 <isdn>

  <!-- Call Control groups
    - A group can contain one interface.  When a group contains more
      than one interface this is nfasGroup
   -->
  <groups>
   <group ID="XYZspan1"
```

```
            termination="TERMINAL"
            switchVariant="EuroISDN"
            BChannelNegotiation="EXCLUSIVE">
        <member interfaceID="XYZspan1">
          <BChannels>
            <BChannelRange firstChannel="1" lastChannel="30">
              <resourcesGroup>default</resourcesGroup>
            </BChannelRange>
          </BChannels>
        </member>
      </group>

    </groups>
    <!-- End of groups -->

  </isdn>
  <!-- End if ISDN -->

 </callControl>
 <!-- End of call Control -->

</pstnConfig>
```

Example 3: 2 T1s configured in NFAS

```
<?xml version="1.0" encoding="utf-8"?>
<!-- RelaxNG shema=pstn-config.rng -->
<pstnConfig version="1.0">

 <!-- List all PSTN interface providers -->
 <ifProviders>
  <sangoma>

   <!-- General parameters specific to Sangoma -->
   <BChannelsThreads count="4"
                priority="TIME_CRITICAL"/>
   <DChannelsThreads count="1"
                priority="HIGHEST"/>
  </sangoma>
 </ifProviders>

 <!-- All PSTN interfaces -->
 <interfaces>
  <sangoma id="XYZspan1"
      wanpipe="1"
      mediaType="T1"
      framing="ESF"
      lineEncoding="B8ZS"
      LBO="0dB"
      clocking="TERMINAL"
      loopbackMode="DISABLED">
    <BChannels defaultPcmLaw="PCMU"
          voicePacketLengthInMs="20">
     <VoiceQualityEnhancement>
      <EchoCancellation mode="NORMAL"
                  comfortNoiseMode="NORMAL"
                  tailDisplacementInMs="0"
                  doubleTalkBehavior="OPTIMAL"/>
      <AcousticEchoCancellation mode="NORMAL"/>
      <LevelControl direction="ALL"
                mode="AUTOMATIC"
                targetLevelIndBM0="-20"/>
      <BackgroundNoiseAttenuation direction="FROM_PSTN"
                        mode="ENABLED"
                        attenuationIndB="-18" />
      <DTMFRemoval direction="FROM_PSTN"
                mode="DISABLED"/>
```

```
    </VoiceQualityEnhancement>
   </BChannels>
   <DChannel channelIndex="24"
          mtuSize="2048"/>
  </sangoma>
  <sangoma id="XYZspan2"
      wanpipe="2"
      mediaType="T1"
      framing="ESF"
      lineEncoding="B8ZS"
      LBO="0dB"
      clocking="TERMINAL"
      loopbackMode="DISABLED">
   <BChannels defaultPcmLaw="PCMU"
          voicePacketLengthInMs="20">
    <VoiceQualityEnhancement>
     <EchoCancellation mode="NORMAL"
                comfortNoiseMode="NORMAL"
                tailDisplacementInMs="0"
                doubleTalkBehavior="OPTIMAL"/>
     <AcousticEchoCancellation mode="NORMAL"/>
     <LevelControl direction="ALL"
                mode="AUTOMATIC"
                targetLevelIndBM0="-20"/>
     <BackgroundNoiseAttenuation direction="FROM_PSTN"
                          mode="ENABLED"
                          attenuationIndB="-18" />
     <DTMFRemoval direction="FROM_PSTN"
                mode="DISABLED"/>
    </VoiceQualityEnhancement>
   </BChannels>
   <!-- No D-Channel required -->
  </sangoma>
</interfaces>

<!-- Call control parameters -->
<callControl>
 <!-- Resources groups

  - Resources groups are used for inbound and outbound BChannel reservation.
  - Resources groups contains a list of BChannels.  To add a range of BChannel
    in a given resources group specify it when the BChannel range is specified
    in a ISDN group or ISDN NFAS group.
  - A given BChannel channel can appear in multiple resources group.  But as
    soon as that B-Channel is used it becomes unavailable in every resources
    groups until it is released.
```

```
-->
<resourcesGroups>

 <resourcesGroup ID="default"
                 direction="BIDIR"
                 outboundHuntingScheme="REVERSE_LINEAR"
                 inboundHuntingScheme="REVERSE_LINEAR"/>

</resourcesGroups>
<!-- End of Resources groups -->

<!-- ISDN configuration

 - Contains the ISDN parameters for each ISDN interface (group) or group of
   multple interfaces (nfasGroup).
 -->
<isdn>

 <!-- Call Control groups
    - A group can contain one interface.  When a group contains more
      than one interface this is a nfasGroup
  -->
 <nfasGroups>
  <nfasGroup ID="Group A"
          termination="TERMINAL"
          switchVariant="NI2"
          BChannelNegotiation="EXCLUSIVE">
    <members>
     <!-- This member will carry the ISDN signaling for all interfaces within
          this NFAS group -->
     <!-- nfasInterfaceID attribute shall be unique for each member within
          a NFAS group -->
     <member interfaceID="XYZspan1" nfasInterfaceID="0" DChannelMode="PRIMARY">
      <BChannels>
       <!-- Note the last channel is 23 because channel 24 is used to carry
            ISDN signaling -->
       <BChannelRange firstChannel="1" lastChannel="23">
        <resourcesGroup>default</resourcesGroup>
       </BChannelRange>
      </BChannels>
     </member>
     <member interfaceID="XYZspan2" nfasInterfaceID="1" DChannelMode="DISABLED">
      <BChannels>
       <!-- Note last channel is 24 here because this channel is not used
            to carry ISDN signaling -->
```

```
            <BChannelRange firstChannel="1" lastChannel="24">
              <resourcesGroup>default</resourcesGroup>
            </BChannelRange>
          </BChannels>
        </member>
      </members>
    </group>

  </nfasGroups>
  <!-- End of NFAS groups -->

 </isdn>
 <!-- End if ISDN -->

</callControl>
<!-- End of call Control -->

</pstnConfig>
```

Appendix F: Modify the Microsoft Windows driver signing options

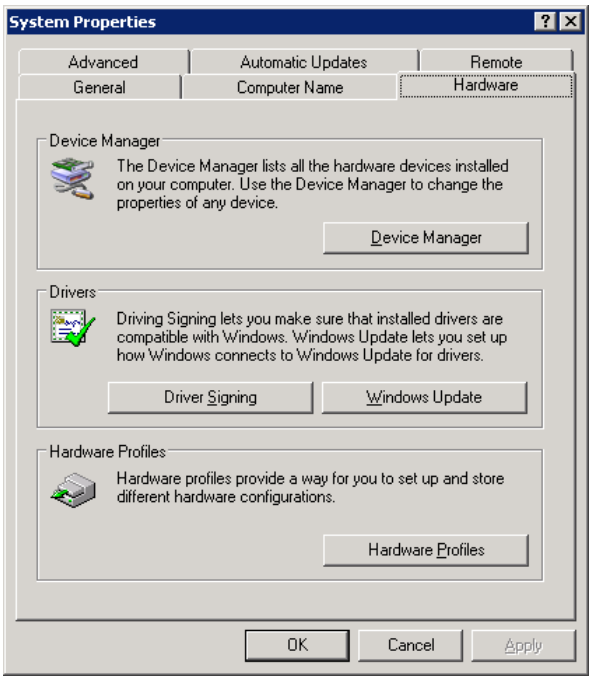
This appendix contains a procedure describing how to change the Microsoft Windows Driver signing options to prevent annoying prompts during Sangoma device drivers installation.

For more detailed information on the Sangoma device driver installation, see the following topics located elsewhere in this guide:

- Installing the device drivers

To disable driver signing prompt:

1. Open the System properties window. To open the system propertie from **Start** Menu, select **Control Panel > System**.



2. In System Properties window, select the **Hardware** tab and click on the **Driver Signing** button.
3. In the Hardware Properties window, select the option **"Ignore - Install software anyway and don't ask for my approval"** and click on OK.