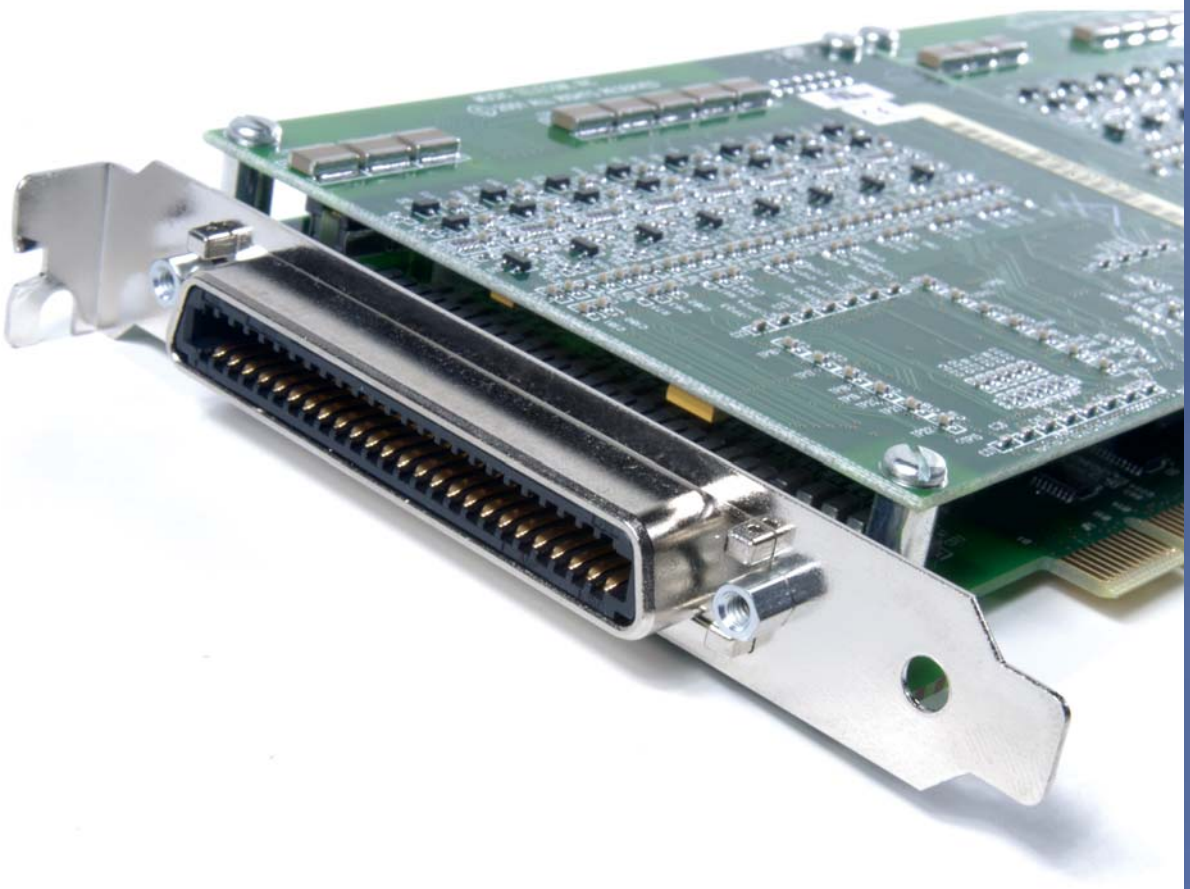
AudioCodes

BLADES BUSINESS LINE



NGX Integration Guide

v. 5.2.0

AudioCodes USA
www.audiocodes.com/blades
27 World's Fair Drive, NJ · 08873
T: 732-469-0880 · F: 732-469-2298

404-2002-001 · Build 090828.01 REVB

Table Of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Chapter 1

Introduction

NGX Overview

Call logging applications are developed for the purpose of recording and archiving phone conversations. Where this meets the demand of most market sectors, call center monitoring applications are significantly more complex. Here the developer must design an application capable of monitoring the call agent's behavior while recording the conversation. Agent behavior is determined by capturing their use of the phone. Loggers use a hardware component that decodes the data passing between the PBX and local phones. The SmartWORKS NGX was designed as a solution for extension tapping.

The NGX provides all features required for call recording: gain control, activity detectors, DTMF detection, Caller ID etc.... The NGX can also tap into the D-channel and decode all PBX/phone communications. This book highlights the use of the NGX when used for call recording. The remaining chapters are used to help a developer use the NGX to integrate with PBXs and obtain D-channel data.

Chapter Descriptions

This book explains how to integrate the SmartWORKS NGX with proprietary PBX networks. Each chapter is described below:

- About the NGX - describes the capabilities of the SmartWORKS NGX applied to standard call recording scenarios. This describes the capabilities of the NGX when D-channel information is not available.
- PBX Integration - provides an overview of implementing the SmartWORKS NGX as a tool that taps the proprietary D-channel passing from PBX to Phone.

NOTE: At the time of this writing, many PBXs are not listed in this book. Please be patient while AudioCodes continues with this documentation project.

- PBX Specific Chapters - one chapter is allocated for each PBX that can be tapped for D-channel by the NGX. Explains specific behavior observed by AudioCodes when testing the PBX, and provides call scenario and phone map information
- Troubleshooting - provides corrective action and troubleshooting tips when using the SmartWORKS NGX

Related Documents

For additional information, refer to the following documents located on the product CD-ROM:

- The *SmartWORKS Developer's Guide* - introduction to the SmartWORK SDK
- The SmartWORKS Function Reference Library - prototype examples of each function in the SmartWORKS SDK
- The *SmartWORKS User's Guide* - getting started with each AudioCodes board
- The *SmartWORKS Utilities Guide* - use of all AudioCodes product utilities
- The *NGX Quick Install* - brief installation and troubleshooting instructions

Document Version Control

The following has been added to this document since the last release:

TABLE 1: VERSION CONTROL

Page	Description
REV A	
	no changes to documentation

Legal Notice

Contacting AudioCodes USA

Your feedback is important to maintain and improve the quality of our products. Use the information below to request technical assistance, make general inquiries, or to provide comments.

TECHNICAL SUPPORT

For programming, installation, or configuration assistance, use the following contact methods:

- Call technical support at 732.469.0880 or call toll free in the USA at 800.648.3647.
- E-mail technical support at blade-support@audiocodes.com. Be sure to include a detailed description of the problem along with PC configuration, AudioCodes hardware, driver versions, firmware versions, a sample program that demonstrates the issue, and any other pertinent information.

SALES AND GENERAL INFORMATION

For sales and general information, use the following contact methods:

- Call us at 732.469.0880 or toll free from the USA at 800.648.3647.
- Fax us at 732.469.2298.
- E-mail us at bladesinfo@audiocodes.com.
- Visit our web site at www.audiocodes.com/blades.

MAILING ADDRESS—USA

Ship packages or send certified mail to us at the following address:

AudioCodes USA, Inc.

27 World's Fair Drive

Somerset, NJ 08873

Chapter 2

About the SmartWORKS NGX

The SmartWORKS NGX hosts 8 high/low impedance digital interfaces, a voice processing subsystem for each interface, an audio jack, and a PCI host interface.

At the center of the SmartWORKS NGX are its DSPs, also called *voice resources*. One voice resource provides enough power to process 8 isochronous channels. Each voice resource has a full set of detectors, encoders, and decoders along with numerous voice processing functions such as automatic gain control (AGC), automatic volume control (AVC), and adaptive echo cancellation. In addition to traditional voice resources, the NGX contains a summation resource. This summation resource has 16 inputs with independent gain control per input, and a 16 channel mixer followed by AVC. All features may be enabled or disabled using SmartWORKS API functions.

The SmartWORKS NGX is an 8 channel board that is field upgradeable to a maximum of 24 channels via two 8 channel MX80 expansion cards. Adding an MX80 expansion card to the NGX provides a cost-effective solution for all system sizes.

Unique Features

The SmartWORKS NGX has been designed to provide a complete call recording solution. The following section outlines board capabilities.

Inputs

Each channel has two inputs: a primary input (NT data) and a secondary input(TE data) which facilitates digital recording. As a result, a two-way mixer is available for each channel on the SmartWORKS NGX board. By default, the mixer is enabled upon a successful load of the DLL. When mixing is disabled, the user application will only receive data from the primary input (incoming side of the conversation).

Each input has its own set of DTMF, MF, and activity detectors (Both MF R1 and R2 tones are detected). The primary input also has Caller ID detection and a Call Progress Monitoring capabilities. All detectors can be enabled or disabled via the SmartWORKS API.

Call Progress Monitoring

SmartWORKS Call Progress Monitoring (CPM) tracks the progress of outbound calls by identifying possible call results (or call states). SmartWORKS CPM was designed around a signal processing service able to detect and recognize of a wide range of Call Progress Signals - such as dial tone, busy signal, or ringback. Each channel on all SmartWORKS boards has been pre-configured with profiles defining typical call progress signals used in North America.

When a PBX produces CPM signals, these are usually proprietary. In order to detect the signals generated by a PBX, the signal profiles must be modified to match the cadence and frequencies of the signals generated by the PBX. For more information refer to the application notes: *Call Progress Monitoring* and *Understanding Signal Profiles*.

Caller ID

All SmartWORKS boards detect Caller ID when it is passed over voice channels (supports Bell 202 and V.23 standards). However, Caller ID is usually not passed over voice channels on digital networks, specifically proprietary network. On some networks, CallerID can be obtained via D-channel information.

BRI Systems

The NGX can be used on ISDN BRI systems. The following call states are reported: alerting, abandoned, connected, held, rejected, released, resumed, retrieved, suspended. When using the NGX to tap ISDN BRI systems, Caller ID is passed to the user application via a call control structure (MT_CC_CALL_INFO).

D-Channel

On some PBX systems Caller ID is passed over the D-channel and displayed on the phone's LCD. Each time the LCD display is updated by the PBX, this information is decoded and the event EVT_MESSAGE_CHANGE is generated. The data on the LCD is passed into a buffer and available to the user application. It is important to note, that some PBXs present the exact CallerID string while some present the caller's name, or extension number. This depends on the PBX model as well as the specific configuration of the PBX in the field.

Activity Detection

The Activity Detector is capable of monitoring line activity/silence and features programmable parameters such as activity threshold, silence threshold, minimum silence, and minimum activity duration. The events generated by this feature can be disabled using the ***MTSetEventFilters()*** API.

Automatic Gain Control

The SmartWORKS NGX offers optional automatic gain control (AGC) followed by a gain stage. The NGX's AGC handles a wide, dynamic range that is typically encountered when a voice logger is connected close to a PBX. By default this feature is disabled, but can be enabled using the ***MTChInputAGCControl()*** API.

Automatic Volume Control

The SmartWORKS NGX has the capability to take the output of a channel mixer, apply automatic volume control (AVC) to it, and adjust the output by programming volume control. The resulted voice stream must be connected to a selected timeslot on the global TDM bus. This feature is controlled by invoking ***MTAVCControl()*** API.

Tone Generation

All SmartWORKS boards are capable of tone generation. When using the SmartWORKS NGX tones cannot be played directly onto the channel and heard by the caller. Tones can be played out onto the CT Bus timeslot with the audio signal. Use ***MTPlayTone()*** to control this option.

Encoders / Decoders

The SmartWORKS NGX offers a wide range of voice encoders and decoders. Digitization methods are programmable on a per channel basis. You may select a GSM 6.10 decoder and a 32 kbp/s G.726 encoder to be used at the same time on any given channel. All codecs supported by the SmartWORKS boards are listed in the *SmartWORKS User's Guide* and the *SmartWORKS Developers Guide*. Recordings cannot be played directly onto a passive NGX channel, however they can be played out onto a timeslot.

TDM Connections

The SmartWORKS NGX can be configured to connect to the global CT bus, MVIP or H.100. This can be used to live monitor a channel. Initially, all channels are configured with both the primary and secondary inputs connected to the channel's DSP resources.

Making a Connection to the CT Bus

The API offers many functions to connect the channel output/input to the CT bus. The following functions are recommended for use with the SmartWORKS NGX:

- ***MTSetOutput()***
- ***MTResetOutput()***
- ***MTSetInputs()***
- ***MTResetInputs()***
- ***MTSetFramerOutput()***

When a channel is opened, but its output is not connected, use ***MTSetOutput()*** or ***MTSetFramerOutput()*** to connect that channel to a specified CT timeslot. When ***MTSetOutput()*** is used, incoming data is passed through the DSPs before routed to the CT Bus. Therefore Gain, AGC, and detection capabilities still apply. If ***MTSetFramerOutput()*** is used, the data is to routed from the output of the framer to the CT Bus while still passing the same input data to the DSPs by default. The data routed to the CT Bus directly from the framers is not processed by the DSPs, therefore no gain control is applied. Refer to the *SmartWORKS Developer's Guide* for more information about each API.

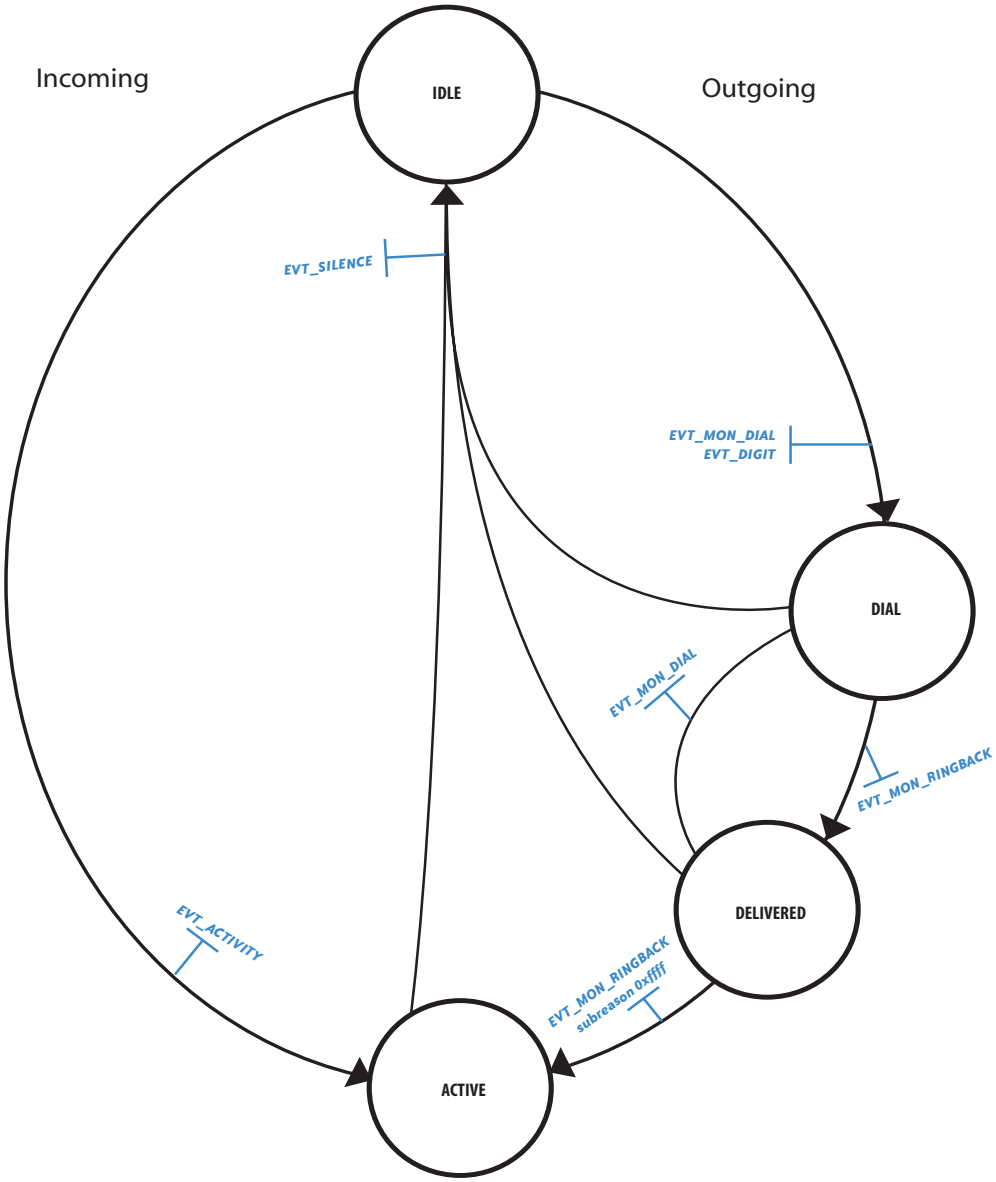
NOTE: Although MTSetCTRoute() is part of the SmartWORKS API, please note that it is not supported by the SmartWORKS NGX.

State Machine

Knowing when to start and stop the recording process is the most important aspect of call logging. If recording begins too soon, dead air is recorded. If it begins too late, the recording is truncated. The NGX has been designed to tap the D-channel, where PBX/Phone communications are passing. These D-channel events are then used to monitor line conditions to determine when to trigger call recording. At the time of this writing, the SmartWORKS NGX has not been developed to decode the D-channel of all PBX models. When working on a system where D-channel data is not available, the call logger must rely on Activity/Silence and Call Progress Monitoring events to determine line conditions.

The following illustrations shows a call state machine that has been developed from events generated on a system where D-channel events are not available. The following chapter, PBX Integration, shows a state machine that can be created when D-channel events are available.

Call State Machine -Vox and CPM only



Error Checking

The SmartWORKS NGX is capable of reporting framer and signal errors. The following APIs can be used to monitor line health.

Framer Statistics

The API ***MTGetNGXFramerStatisitc()*** retrieves network interface statistics per framer:

- a count of PBX signal errors
- a count of Phone signal errors
- a count of synchronization loss errors
- amplitude of the PBX signal in volts
- amplitude of the phone signal in volts
- amplitude of the noise level in volts
- clipping status (indicating the incoming voltage is too high)

Where any field displays a total count of errors, this equals the number of errors accumulated on the specified framer either since the system was started or since the last call of **MTClearFramerStatistic()**.

Loss of Signal Errors

Using SmartWORKS APIs, user can also monitor Loss of Signal alarms, reported by the framer. The API ***MTArmFramerAlarm()*** must be used to enable this alarm. Once armed, the event EVT_LOS_ALARM is generated each time a loss of signal occurs. The API ***MTGetFramerAlarmStatus()*** can also be used to obtain status. Any non-zero value indicates a loss of signal. Once an error is detected the framer alarm must be re-armed.

Chapter 3

PBX Integration

The SmartWORKS NGX fits into call centers for the passive tapping of phone conversations behind proprietary PBXs. With the NGX, call recorders can track agents sitting at multiple locations throughout a call center. To accomplish this, the NGX taps into the data channel (D-channel) and decodes the communication passing between the PBX and phones.

This section shows how the SmartWORKS NGX is used to capture D-channel events by explaining the following:

- Configuration - configuration required for D-channel tapping after the NGX card has been installed behind the PBX
- PBX Integration - lists each PBX model supported in a VOX or D-channel scenarios, including all phone models tested
- D-channel Events - explains the types of D-channel events reported by the NGX
- Observed Variations with PBX Models- describes how the information passed to the user application varies depending on the PBX model
- Call State Machine - an example of a call state machine that can be built when combining D-channel events with Call Progress Monitoring (CPM) events

Installation and Configuration

Complete installation instructions are available in the *SmartWORKS User's Guide*. The purpose of this section is to explain the installation and configuration required for D-channel tapping.

Cable Lengths

When installing the NGX, users must verify that cables lengths fall within guidelines set by AudioCodes. Three cable lengths are measured:

- the tap to the NGX
- the tap to the phones
- the tap to the PBX

The NGX firmware assumes that the shortest cable is from the tap point to the NGX card. The recommended cable length varies per PBX model. More information is provided in each PBX chapter of this book.

CONFIGURATION

After the NGX card has been installed on the digital network, specific configuration steps are required to generate D-channel events.

NOTE: If any configuration settings are modified, the board must be re-started for the new settings to take effect.

CLOCK SETTINGS

AudioCodes recommends setting the NGX's clock source to the PBX. This avoids any synchronization problems. Use the ***API MTSetCTMasterClock()*** to set the clock source parameter to NET1. The default setting is to local - the board's clock (OSC).

PBX FIRMWARE

Using the Control Panel, select the **Board** tab. The *PBX Type* field must be set. Use the drop-down menu to select the name of the tapped PBX. Click APPLY (this is required!). The firmware used for this PBX will be automatically installed the next time the board is restarted. Repeat this step for each NGX board and daughter card.

TDM ENCODING

Using the Control Panel, select the **Board** tab. The *TDM Encoding* field must be set. Generally speaking, µ-Law is used in North America and Japan, A-Law is used in Europe. This setting is also PBX dependant. It is highly recommended that you check PBX before installing the NGX onto the network.

If this setting is not correct, the recording may sound "metallic".

D-CHANNEL EVENTS

The NGX board does not generate D-channel events by default. This must be enabled. Using the Control Panel select the **Board** tab. The *D-channel* option must be enabled. If the NGX does not support D-channel for a particular PBX model, then this option in the control panel is greyed out.

EVENT UPDATES

Some PBXs send duplicate commands or phone status reports over the line to the phones. This applies to events that indicate a change of state, which includes lights, audio, hook, ring/tone, and display events. As a result, the NGX decodes this information and generates extraneous events. This can be controlled by disabling the *Event Updates* option in the Control Panel (under Board tab). Once this is disabled, all duplicate events are filtered.

NOTE: AudioCodes always tests and documents PBX behavior with Event Updates disabled.

Phone Model Support

The SmartWORKS NGX has been designed to integrate with multiple PBX models. A complete listing of each PBX model, inlcuding phone model support and Dchannel data reported is available on the online support system. Look for the NGX/IPX Support Matrix.

D-Channel Events

All D-channel events are passed to the user application via the MT_EVENT structure. Depending on the exact nature of the event, the Subreason and Xtra Info fields may also be populated with information. If more information is returned to the user the ptrBuffer field is populated with a pointer and the DataLen field which is populated with the data length of the buffer. All events are defined in the *SmartWORKS Function Reference Library* . For a list of the Dchannel events reported per each PBX integration, refer to the NGX/IPX Support Matrix found on the online support system.

Event Types

AudioCodes classifies D-channel events into two categories based on where the event was generated - PBX (NT) or phone (TE). Events are further classified by type:

PBX Events

These events are generated by the PBX. Instructions are passed over the D-channel to the phone as a command to perform some type of action. These events are also referred to as Command events. The following types of command events are generated by the PBX:

- Signaling - these events indicate the PBX is commanding the phone to produce a tone (ringing, or incoming page) (Call Progress Monitoring events are also discussed in this section. CPM events are not D-channel events).
- Audio Events - indicate the PBX is controlling external audio devices such as headsets or microphones
- LEDs - these events correspond to light changes on the phone. Light events are important indications when monitoring call states and feature activity.
- Display - these events indicate that the LCD on the phone has been updated. These are usually related to the clock display, or messages displayed on the LCD.
- Call State - these events are generated with a change in call state

NOTE: Some phones control their own audio, lights, or display information. In this scenario, the action taken by the phone may not be reported to the PBX and as a result, the NGX is unable to report this behavior to the user application.

Phone Events

These events are generated by the phone after an action has been taken (i.e. button pressed). The phone is informing the PBX that something has occurred. Events generated by the phone have been classified by the following types:

- Hook State - off hook and on hook changes occur when the handset is removed or replaced
- Button Depression events - indicate that a button on the phone was used. For example: digits, speaker buttons etc. Button events can include both a pressed or released event, depending on the PBX.

D-Channel Events per PBX

A table listing D-channel events per PBX is available in the customer support section online. After logging into the online support system, go to the Files and Documents section and look for the NGX folder.

Each PBX chapter in this book lists the D-channel events supported per model.

Observed Variations with PBX Models

The information passed to the user application varies depending on the PBX model on the network. AudioCodes takes care to record as much as it can about the behavior of each PBX. The following section lists behaviors important for many call logging applications. This section is not meant to be an exhaustive list of all PBX behaviors, but rather a high level discussion for developers who are getting started. Refer to a specific PBX chapter for information.

PBX Command Events

These are the events passed along the D-channel from the PBX to the phone.

Signaling Events

All application developers rely on signaling information to monitor call states. Most signalling events are managed by the Call Progress Monitoring feature and are not D-channel events.

Dial Tones

When an outgoing call is initiated, some type of tone is played onto the line informing the caller that a line has been captured. This tone is typically a standard dial tone. If the PBX generates a dial tone that is played over the line the NGX's Call Progress Monitoring feature detects the signal and generates the EVT_MON_DIAL event. This is not a D-channel event. Some PBXs pass a D-channel command to the phone which generates a dial tone. This D-channel event is not decoded by the NGX.
NOTE: Channels are configured with signal profiles that match standard North American signals. Many signals generated by PBXs are proprietary, and do not generate signals that match the default values. All signal profiles can be modified to work with the local environment. Refer to the application notes: *Call Progress Monitoring* or *Understanding Signal Profiles* for more information.
NOTE: While most systems generate a dial tone, some PBXs generate a proprietary sound, or even music.

Ring Indications

Ringback tones indicate a far phone is ringing when an outbound call has been placed. The Call Progress Monitoring system is able to detect ringback signals and report an EVT_MON_RINGBACK event. This is not a D-channel event.

For incoming calls, most PBXs pass a command to the phone instructing it to ring. These events are D-channel events.

Ringback Tones

To detect a phone ringing on the far end, the NGX card relies on the Call Progress Monitoring (CPM) feature. The SmartWORKS CPM was designed around a signal processing service capable of analyzing cadence pattern and frequencies of incoming signals. Profiles are used to store values that define cadence and frequency of each call progress tone. All default tone templates are based on North American values but are programmable to adapt to any network. If the tone generated by the network has a different cadence or frequency then it will not be detected by the CPM system. For more information about Call Progress Monitoring refer to the application notes: *Call Progress Monitoring* or *How to Analyze a Signal*.

Determine Ring Count: The Call Progress Monitoring feature is capable of counting total ring count. A count of each complete cycle of a ringback can be sent to the user application via the subreason field of the MT_EVENT structure. For more information refer to the application note: *Call Progress Monitoring*.

PBX Alerting / Ringing

On some networks, the PBX sends signalling instructions alerting the phone of an incoming call and commanding it to ring. These instructions are decoded as EVT_RING_ON or EVT_RING_OFF events. The number of RING events passed to the phone varies on both the PBX or phone model used on the system.

Determine Ring Count: On some systems, for each complete cycle of a ring tone one unique EVT_RING_ON and EVT_RING_OFF event is generated. In this case the user application can count the total number of events and determine the number of times the phone rang. On other PBX networks the EVT_RING_ON is generated only once. When the ring tone is no longer detected the EVT_RING_OFF event is generated. In this scenario, the timestamps of the two events are used to measure the length of time the phone was ringing.

For information on specific PBX behavior refer to each PBX chapter in this book.

Audio Change Events

These events report the current audio state. In some cases, the phone model supports it's own audio and the PBX does not control it. If the phone does not report this action to the PBX, then the NGX is unable to report the change in audio state to the user application.

When this event (EVT_AUDIO_CHANGE) is reported the subreason field indicates what device is being managed and it's current state. This field is a 32 bit field. Bits 4-31 are reserved. Transmit and receive are always in respect to the phone's position. Transmit - phone to PBX, and Receive - PBX to phone. The following table lists all possible options. Not all states are observed with each PBX

.

TABLE 1: EVT_AUDIO_CHANGE BIT OPTIONS

Device State	SPKR RECV	SPKR TRANS	HDSET RECV	HDSET TRANS (LSB)	HEX VALUE
All devices are off	0	0	0	0	0x0000
Handset transmitting	0	0	0	1	0x0001
Handset receiving	0	0	1	0	0x0002
Handset active (Rx/Tx)	0	0	1	1	0x0003
Speaker transmitting	0	1	0	0	0x0004
Handset/speaker transmitting	0	1	0	1	0x0005
Speaker transmitting Handset receiving	0	1	1	0	0x0006
Speaker/handset transmitting Handset receiving	0	1	1	1	0x0007
Speaker receiving	1	0	0	0	0x0008
Handset transmitting Speaker receiving	1	0	0	1	0x0009
Handset/speaker receiving	1	0	1	0	0x000A
Handset transmitting Speaker/handset receiving	1	0	1	1	0x000B
Speaker transmitting and receiving	1	1	0	0	0x000C
Handset/speaker transmitting Speaker receiving	1	1	0	1	0x000D
Speaker transmitting Handset/speaker receiving	1	1	1	0	0x000E
All devices are active	1	1	1	1	0x000F

LCD DISPLAY EVENTS

Typically, the PBX commands the phone to update the display on the phone's LCD. As the display command is passed along the D-channel, the NGX decodes the command and reports the information to the user application. Five display events are available:

- EVT_MESSAGE_CHANGE - the PBX passes information to the phone so that it can be displayed on the phone's LCD. On many PBXs, the callerID is passed along the D-channel with this method. The ptrBuffer field of the MT_EVENT structure points to the buffer which holds the information passed to the phone's LCD. The "datalength" field is populated with the size of the data buffer pointed to by ptrBuffer, including a null character terminator at the end.
- EVT_DISPLAY_MESSAGE - the PBX commands the phone to display a message that is stored in the phone's memory. The PBX does not send the message data, but it sends the ID of the message that should be displayed. The message ID is passed to the user application in the subreason field event structure. Any messages that have been observed by AudioCodes during testing are documented in each PBX chapter in this book.
- EVT_DISPLAY_CLOCK - the PBX commands the phone to update the clock display
- EVT_DISPLAY_TIMER - a timer is displayed when a call is connected. This timer usually measures call duration.
- EVT_DISPLAY_CLEAR - the command to clear the LCD display on the phone.

NOTE: The phone may also change the LCD in response to a user's action (button pressed). This information is not passed along the D-channel and is missed by the NGX.

Agent ID / Caller ID

Refer to the section above that explains LCD display. On most PBX networks caller ID and agent ID are not present in the B-channel. Most PBX models pass caller and agent ID in the D-channel when the command is passed to update the phone's LCD. The NGX decodes this command and reports a EVT_MESSAGE_CHANGE event. The data displayed on the LCD is saved into a buffer and available for the user application in ASCII format. The agent ID and callerID must be parsed from the string. For information as to whether the agent or caller ID information can be obtained via D-channel refer to each PBX chapter in this book.

LED/Light Events

Monitoring lights is an excellent way of tracking call states and feature activity when working on a digital system. By paying attention to light events, application developers can track whether the call is placed on hold, transferred or disconnected. On most networks, the PBX controls phone lights and the NGX is capable of decoding the light commands and reporting this to the user application.

On many phones the light is mapped to a specific function. For example, some phones have Hold or Speaker buttons. When these features are in use then a corresponding light is illuminated and EVT_HOLD_LIGHT_ON/FLASHING or EVT_SPEAKER_LIGHT_ON/FLASHING can be reported.

In other cases, some lights correspond to a programmable function button. In this case an EVT_FUNCTION_LIGHT_ON/FLASHING is reported. **NOTE:** EVT_FUNCTION_LIGHT_ maps to EVT_LIGHT_ events for backwards compatibility.

The light subreason field indicates the light number and color. Represented as a hex value the following holds true 0xRRRRCCNN where R = reserved, C = color, and N = light number. The following table represents each bit value of the subreason field:

RRRR	CC				NN
b31-b16	b15-b11	b10	b9	b8	b7-b0
reserved	reserved	Amber	Red	Green	Light Number

NOTE: These may vary depending on phone model and PBX. Refer to a specific PBX chapter for more information.

Alcatel phones do not have lights, instead pictures are presented on the phone display. The following bits are set:
Bits 0-7 = "Light" Number
Bits 8 = square, located on the right side of the display area
Bit 9 = music symbol
Bit 10 = square, located in the center of the display area
Bit 11 = handset symbol
Bit 12 = square, located on the left side of the display area
Bit 13 = alert symbol
Bits 14=31 Reserved

Light Cadence

On some networks, the light is simply turned ON/OFF. Here, the EVT_HOLD_LIGHT_ON/OFF event or EVT_FUNCTION_LIGHT_ON/OFF events are reported respectively.

On other networks, the lights may flash or wink. The following cadence patterns are reported to the user application:

EVT_XXX_LIGHT_FASTFLASHING
EVT_XXX_LIGHT_FLASHING
EVT_XXX_LIGHT_OFF
EVT_XXX_LIGHT_ON
EVT_XXX_LIGHT_QUICKFLASH
EVT_XXX_LIGHT_VERY_FASTFLASHING
EVT_XXX_LIGHT_WINK
EVT_XXX_LIGHT_SLOW_WINK
EVT_XXX_LIGHT_MEDIUM_WINK

Cadence patterns vary per PBX manufacturer. Refer to a specific PBX chapter for more information.

Call State Events

These events are PBX specific and are used to report a change in call state. Refer to the individual PBX chapters in this book for more information.

Phone Action Events

These events are reported when the phone passes information to the PBX.

OFF_HOOK/ON_HOOK events

Most phones have cradles for handsets. When these are pickup or placed down, the corresponding events EVT_ON_HOOK and EVT_OFF_HOOK are reported to the user application.

However, in many call centers, agents are using headsets therefore OFF_HOOK/ON_HOOK events are not reported. Here the application developer must design their system to monitor for other line conditions. Each PBX network is unique. A few examples are provided below. **NOTE:** This is not an exhaustive list. Refer to a specific PBX chapter for more information:.

EVT_AUDIO_CHANGE - this is reported when the audio state of a microphone, speaker or handset changes. When no OFF_HOOK or ON_HOOK events are reported this can be used to monitor the start or stop of a call.

EVT_LINE_BUTTON_PRESSED - on some phone models, this is used to answer an incoming call or seize a line to initiate an outgoing call

EVT_RELEASE_BUTTON_PRESSED - on some phone models this button is used to disconnect an active call.

Button Depression Events

Whenever possible, the NGX identifies each button by the type of action taken when it is used. For example, EVT_HOLD_BUTTON_PRESSED/RELEASED. If a button is not specific, or it is a programmable button, then the event EVT_FUNCTION_BUTTON_PRESSED/RELEASED is used. There are many types of button events. AudioCodes organizes them based on purpose:

- Line - line button events may be reported is a specific button is used to answer a call or disconnect an active call
- Soft - these buttons are used around the phone's LCD when an agent is navigating the menu displayed on the LCD.
- Fixed - these buttons serve a fixed purpose such as Hold, or Speaker features
- Feature/Function - these buttons are programmable and can also be referred to as generic buttons
- Digits - the EVT_DIGIT_PRESSED is generated when a digit button is used

Action Missed by the NGX

At times some action performed on the phone is not reported to the PBX. As a result, nothing is passed over the line in the D-channel and this action is not detected by the NGX. Some examples of missed behavior are as follows:

- Volume control - volume control set on the phone may not be reported to the PBX.
- "hot mic" - where the microphone is turned on by the PBX, but never turned off between phone calls.
- canned messages - phones may be configured with programmable messages, each with an unique ID. As a result, when the command to display information is passed from the PBX to the phone only the message ID is passed over. The actual message is not sent from the PBX to the phone.

- audio changes - typically the PBX commands the phone to turn the audio on or off. In some scenarios the phones control the audio and do not inform the PBX that the audio state has changed
- lights - when a button is depressed on the phone, the phone may control the light associated with this button. In this case the PBX does not need to command the phone to turn the light on/off and the NGX is unable to report this action

It is important for application developers to take note of these variations. AudioCodes does it's best to document this information. Any noted behaviors are listed in the chapter corresponding with each PBX model.

CRC Error Checking

Many PBXs provide a checksum or redundancy check for errors in the protocol. When CRC error checking is used, errors are posted to the system event viewer on a per packet basis. If errors are present, the NGX does not decode the corresponding D-channel reducing the number of invalid D-channel events. Some PBXs do not follow this procedure and, as a result, invalid data can be captured by the NGX and provided to the user.

CRC errors usually indicate poor line conditions or improper wiring. It is important to also monitor line conditions by using the error checking APIs: ***MTGetFramerStatus()*** and ***MTGetNGXFramerStatistic()***.

For information on whether a particular PBX model supports CRC error checking refer to each PBX chapter in this book.

Dialed Numbers (DTMF) Detection

When tapping any digital network, there are two ways to detect a DTMF keystroke:

- Standard DTMF detectors - the NGX card uses on-board detectors to identify in-band DTMF tones as they pass over the line. Both incoming and outgoing DTMF signals can be detected by the NGX. EVT_DIGIT is generated. The subreason field includes the tone number that has been detected (0-9, *, #) and the XtraInfo field shows the direction - 0 (NT) or 1 (TE).
- EVT_DIGIT_PRESSED - as a digit is pressed on the phone, the phone alerts the PBX and the data is tapped by the NGX. EVT_DIGIT_PRESSED is generated. The subreason field includes the decoded digit value (0-9, #, *, or any integrated keyboard character).

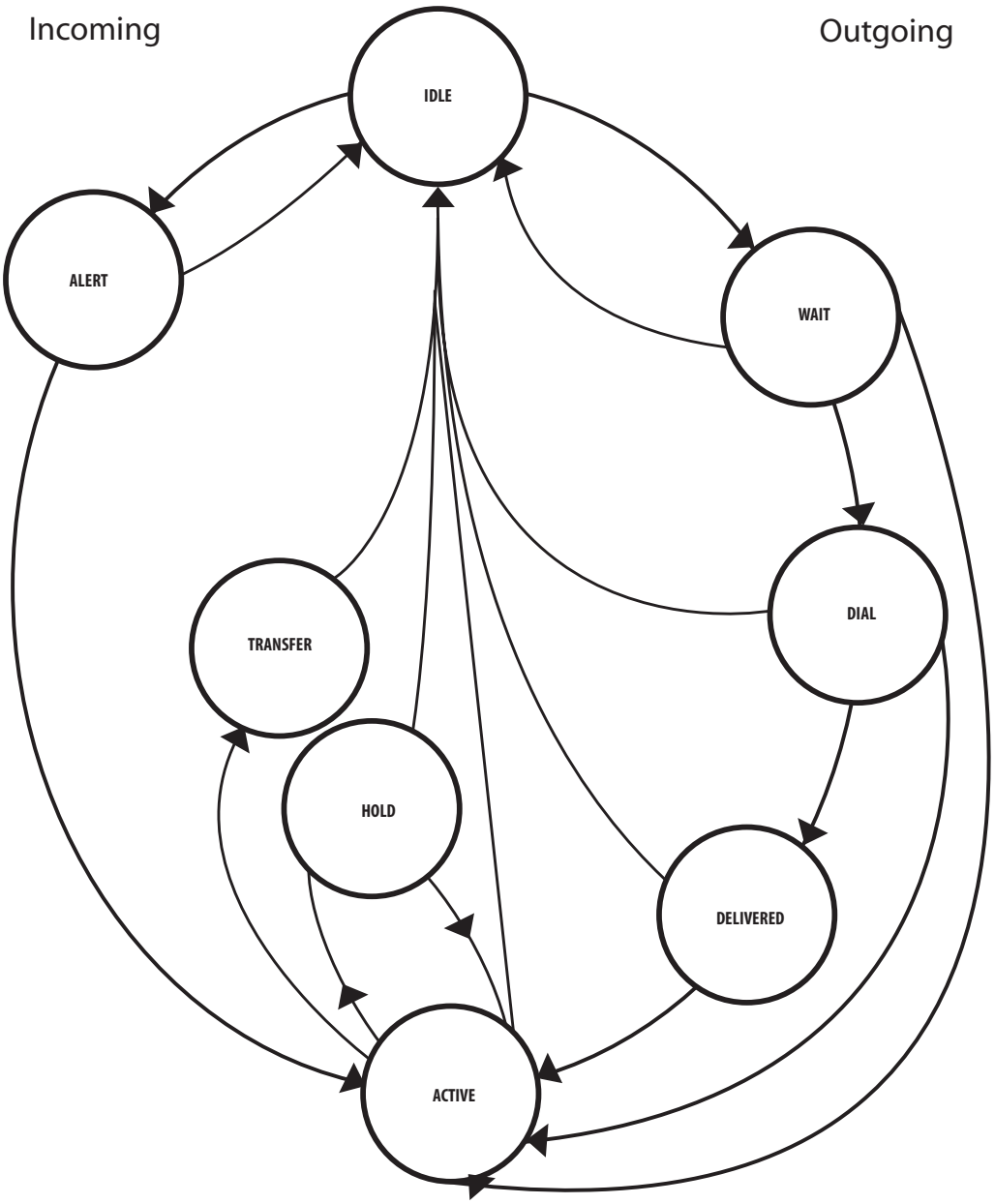
Call State Machine with D-Channel Events

The ability to decoded the data passing along the D-channel gives call loggers an advantage. As PBX/phone communications are decoded they are passed up to the user application. The application developer, relying on this data, can build a detailed call state machine used to track the progress of each call on the system. This information can be used to determine when to stop/start call recording, plus provide user applications the information required to design a complex call monitoring system.

The previous chapter shows an example of a state machine when D-channel events are not available. This diagram is provided on the next page so that is can be compared to the diagram which illustrates a call state machine designed around D-channel events.

D-channel events are used to determine the cause in the change of call state. Since each PBX network is unqiue the diagram does not show them. Refer to the table on the preceeding pages for information about the types of events reported prior to a change in call state.

Call State Machine - when D-Channel is available



The events for each channel are not presented to the user application with a session number. As each channel can have several sessions and each session may be in a different state it is recommended that the user application supply a session number. This is important when tracking calls that are transferred.

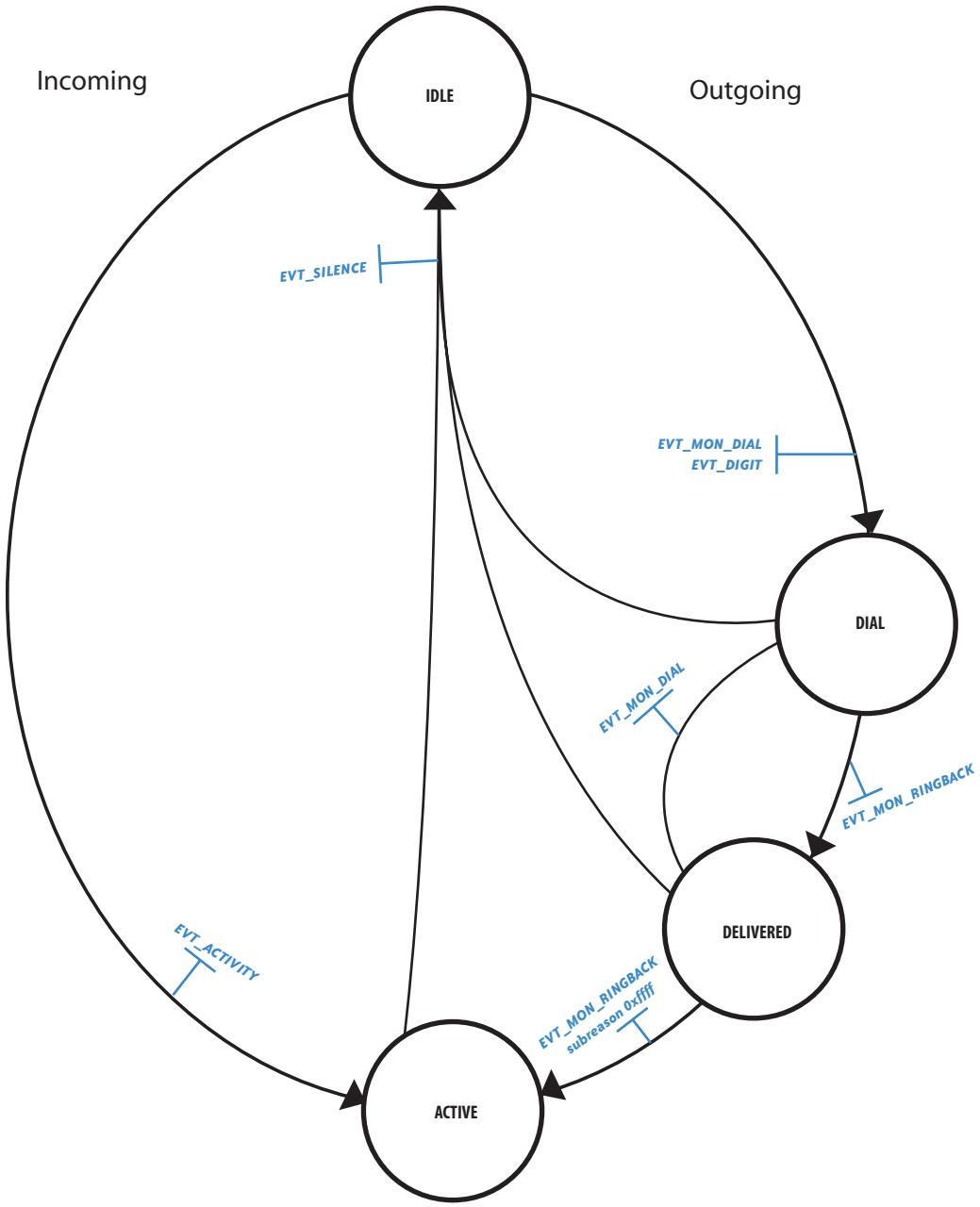
The application developer should take time to become familiar with the types of events produced by a specific PBX when a call moves into another state. The table below lists each call state and provides a description. The application developer must take the time to learn the types of events than can be generated on a particular network when the call state changes.

NOTE: Each PBX model is unique. The type of D-channel events vary from PBX manufacturer, PBX configuration as well as the phone model used. This table should be used only as a quick reference guide and is not recommended for development. The application developer must observe the behavior of each PBX to design a call state specific to their network.

Call State Begin State --> End State	Description
Outgoing Calls	
Idle --> Wait	**REQUEST** Agent initiates phone call. Awaiting acknowledgement from the network indicating that a line has been seized(dial tone).
Wait --> Idle	**ABANDONED** The agent abandons the call the network has acknowledge that a line has been seized (dial tone).
Wait --> Dial	**ACKNOWLEDGED** Dial tone is present on the line indicating that a line has been seized. Agent dials the called number.
Wait --> Active	**CONNECTED** Glare condition. An incoming call was present on the line when the agent initiated the call.
Dial --> Delivered	**Far Side Acknowledgment** The network is alerting the far side that at call is present. Ringback tones are present on the line.
Dial --> Idle	**ABANDONED** The agent abandons the call before it is connected on the far end.
Dial --> Active	**CONNECTED** Glare condition. An incoming call was present on the line when the agent initiated the call and began dialing.
Delivered --> Idle	**ABANDONED** The agent abandons the call before the call is delivered. The local CO may have delivered a SIT tone or the far side may have rejected the call.

Call State Begin State --> End State	Description
Delivered -- > Active	**CONNECTED** Call is connected. Ringback has stopped.
Active --> Idle	**RELEASED** The call is released
Active --> Hold	**HELD** The call is placed on hold.
Active --> Transfer	The call is transferred.
Hold -- > Idle	**ABANDONED** **RELEASED** The call is abandoned if the party on the far end hangs up. It is released if the agent terminates the call.
Hold -- > Active	**RETREIVED** The call on hold is retrieved and returns to the Active state.
Transfer --> Idle	**ABANDONED** **RELEASED** The call is abandoned if the party on the far end hangs up. It is released if the agent terminates the call.
Incoming Calls	
Idle --> Alert	**ALERTING** The phone is notified of an incoming call.
Alert --> Idle	**ABANDONED** **REJECTED** The call is abandoned at the far end, or rejected locally.
Alert --> Active	**CONNECTED** The call is connected.

Call State Machine -Vox and CPM only



Chapter 4

Alcatel

This chapter highlights the use of the SmartWORKS NGX when tapping the Alcatel PBX.

NOTE: All data in this chapter explains the expected behavior when tapping the Alcatel PBXs - Alcatel 2400 and the Alcatel OmniPCX 4400 with the UA16 line card (software version is not documented). If another software version or line card is used, different D-channel patterns may be observed.

Phone Model Support

The following table shows the phone models that have been tested in a tapped environment. This information is updated frequently. For an updated copy refer to the Online Support system's Files and Documents section in the PBX_MATRIX folder.:

Model	Vox		D-Channel	
4003	✓	S	✓	S
4011	✓	S	✓	S
4012	✓	S	✓	S
4023	✓	S	✓	S
4034	✓	S	✓	S
4004 Reflex First	✓	T	✓	T
4010 Reflex Easy	✓	T	✓	T
4020 Reflex Premium	✓	T	✓	T
4035 Reflex Advanced	✓	T	✓	T

Status:
T - tested in house
S - supported based on product family (not tested)
R - tested by third party
N - not tested, it may work
W - tested, will not work

Installation and Configuration

Complete installation and wiring diagrams are available in the *SmartWORKS User's Guide*. The following section highlights important installation notes for this particular PBX.

INSTALLATION

This PBX is a two wire (2W) model. Follow the standard 2W installation procedures outlined in the *SmartWORKS User's Guide*.

Cable Lengths

The following are the maximum cable lengths (in feet) recommended by AudioCodes:

PBX to tap - 30*

tap to NGX - 30*

tap to phones - 1500*

* These values have not be tested, but are a fair estimate. AudioCodes customers have deployed the NGX with this PBX, however, we cannot provide exact cable length specifications. Recommended cable lengths are published in the PBX Support folder on the Online Support system. This document is continuously updated as more systems are tested.

Configuration

After the NGX card has been installed on the digital network, specific configuration steps are required to generate D-channel events.

Clock Settings

AudioCodes recommends setting the NGX's clock source to the PBX. This avoids any synchronization problems. Use the ***API MTSetCTMasterClock()*** to set the clock source parameter to NET1. The default setting is to local - the board's clock (OSC).

PBX firmware

Using the Control Panel, select the **Board** tab. The *PBX Type* field must be set. Use the drop-down menu to select the name of the tapped PBX - *Alcatel4200/4400*. Click APPLY (this is required!). The firmware used for this PBX will be automatically installed the next time the board is restarted. Repeat this step for each NGX board and daughter card.

TDM Encoding

Using the Control Panel, select the **Board** tab. The *TDM Encoding* field must be set. Generally speaking, µ-Law is used in North America and Japan, A-Law is used in Europe. This setting is also PBX dependant. It is highly recommended that you check PBX documentation to verify the setting required on your local network.

By default, the Alcatel is set to A-Law. This is a programmable value and AudioCodes recommends checking this setting prior to configuring the NGX.

If this setting is not correct, the recording may sound "metallic".

D-channel Events

The NGX board does not generate D-channel events by default. This must be enabled. Using the Control Panel select the **Board** tab. The *D-channel* option must be enabled.

Event Updates

Many PBXs send duplicate commands or phone status reports over the line to the phones. As a result, the NGX decodes this information and generates extraneous events. This can be controlled by disabling the *Event Updates* option in the Control Panel (under Board tab). Once this is disabled, all duplicate events are filtered.

The Alcatelsends phone status information. If the Event Updates option is disabled in the Control Panel, these events are ignored.

D-Channel Events

The following is a list of all D-channel events reported when tapping Alcatel PBXs. All events have been grouped by event type.

Results vary depending on the configuration of the PBX in the field, along with the phone model used at the customer site. AudioCodes does not guarantee that all events are reported at each PBX site.

PBX Command Events

The following events are reported from commands passing from the PBX to the phones.

Signalling Events

EVT_RING_OFF
EVT_RING_ON

Audio Events

EVT_AUDIO_CHANGE

Call State Events

No call state events are reported.

LED (Light) Events

EVT_FEATURE_LIGHT_FASTFLASHING
EVT_FEATURE_LIGHT_FLASHING
EVT_FEATURE_LIGHT_OFF
EVT_FEATURE_LIGHT_ON
EVT_FUNCTION_LIGHT_FLASHING
EVT_FUNCTION_LIGHT_OFF
EVT_FUNCTION_LIGHT_ON

Display (LCD) Events

EVT_DISPLAY_CLOCK
EVT_MESSAGE_CHANGE

Phone (Action) Commands

The following events are reported from data generated by the phone and passed to the PBX.

Hook State Events

EVT_OFFHOOK
EVT_ONHOOK

Button Depression Events

EVT_DIGIT_PRESSED
EVT_RELEASE_BUTTON_PRESSED
EVT_SHIFT_BUTTON_PRESSED
EVT_SHIFT_BUTTON_RELEASED
EVT_CTRL_BUTTON_PRESSED
EVT_CTRL_BUTTON_RELEASED
EVT_MENU_BUTTON_PRESSED

Alcatel Behavior

Each PBX exhibits unique behaviors. This section shows how common line conditions are handled by the Alcatel. This section is not meant to be an exhaustive list, but rather an overview of some of the behavior observed by AudioCodes.

NOTE: For an overview of the following topics refer to the PBX Integration chapter of this book.

Dialed Numbers (DTMF) Detection

When tapping the Alcatel the DTMF is not passed in-band. Digits can be detected when buttons on the tapped phone are used to dial numbers. This information is decoded from the D-channel and reported as EVT_DIGIT_PRESSED. The exact digit (in ASCII format) is presented to the user application in the subreason field of the MT_EVENT structure.

CallerID

On proprietary PBX networks, callerID is sometimes displayed on phone LCDs. This varies depending on the configuration of the PBX. As a result, AudioCodes cannot guarantee that CallerID is available when tapping the Alcatel.

When the phone's LCD is updated by the PBX for any reason, the NGX decodes this information and the event EVT_MESSAGE_CHANGE is reported. The ptrBuffer field of the MT_EVENT structure points to the buffer and the "datalength" field is populated with the size of the data buffer pointed to by ptrBuffer, including a null character terminator at the end. **NOTE**: The callerID information must be parsed from the string.

CRC Error Checking

The Alcatel relies on a 1-byte CRC check. Invalid messages are not decoded, and error message are logged in the system event viewer.

CRC errors usually indicate poor line conditions or improper wiring. It is important to monitor line conditions by using the error checking APIs: ***MTGetFramerStatus()*** and ***MTGetNGXFramerStatistic()***.

PBX Command Events

The following section highlights the observed variations noted with this particular PBX.

Signalling Events - Call Progress Tones

The Alcatel generates call progress tones and passes them in-band over the line to the phone. As a result, the Call Progress Monitoring feature is able to detect the signal and the event EVT_MON_DIAL,BUSY... is reported. The signal profiles configured by default on each channel are based on North American standards. The profile parameters of the dial tone must be modified to match the cadence and frequency values of the signal produced by the Alcatel. Refer to the application notes: *Understanding Signal Profiles* for more information.

SIGNALLING EVENTS - RING TONES

Alcatel, commands the phone to begin playing a ring tone. When this tone is no longer required, the PBX then issues the command to stop. EVT_RING_ON is reported at the start of play and EVT_RING_OFF when finished. To determine the number of rings, the user application must rely on the timestamp between events to determine how long the phone was ringing before it was answered.

Ring Type

The Alcatel PBX controls the type of ring tone played. When the command is issued to the phone to play a ring tone, the PBX also controls the melody, cadence pattern and type. This information is passed to the user in the subreason field of the MT_EVENT - 0x00mmnnvv.

mm - melody (0-F)
nn - cadence pattern
vv - volume level (1-7)

AUDIO EVENTS - EVT_AUDIO_CHANGE

The Alcatel PBX controls the audio on the phone's handset, speaker or microphone. When the PBX is either enabling/disabling the audio on a device, the NGX reports the EVT_AUDIO_CHANGE event with the subreason field indicating the type of device:

Subreason	Description
0x00	Audio Off
0x0F	Audio On Handset and speaker
0x0D	Audio On Handset and speaker monitor
0x08	Audio On Speaker Monitor
0x0C	Audio On Speaker Phone (speaker and microphone)

With Ring Events

The Alcatel passes audio commands when the ring on/off command is passed to the phone. As a result, when the NGX reports EVT_RING_ON/OFF, a corresponding EVT_AUDIO_CHANGE event is reported. The subreason field of both events indicate the type of device that is being controlled by the PBX.

LCD DISPLAY EVENTS

When decoding Alcatel D-channel, the following LCD events are reported: EVT_MESSAGE_CHANGE, and EVT_DISPLAY_CLOCK.

When EVT_MESSAGE_CHANGE is reported, the ptrBuffer field of the MT_EVENT structure points to the buffer and the "datalength" field is populated with the size of the data buffer pointed to by ptrBuffer, including a null character terminator at the end. **NOTE**: The callerID information, when available, is parsed from the data string.

LED LIGHT EVENTS

The light behavior has not been documented when observing the light events while tapping the Alcatel PBX. Application developers should monitor what events (call on hold, incoming call) map to flashing or fastflashing events.

Subreason field:

The light subreason field indicates the light number and color. Represented as a hex value the following holds true 0xRRRRCCNN where R = reserved, C = color, and N = light number. The phones that integrate with the Alcatel do not change colors. Therefore the color bits are not used. The following illustrations shows the bits values set in the subreason field when decoding the Alcatel:

RRRR	CC	NN
b31-b16	b15-b8	b7-b0
reserved	reserved	Light Number

Some of the Alcatel phone models rely on images rather than lights. These display changes are reported by the NGX as EVT_LIGHT_ events. The subreason field indicates the type of display by settings bits to '1' - enabled.

The following bits are used per each image. When the bit is flagged (set to one) the image is displayed on the phone:

NOTE: The type of image used, and the style of the image may vary per phone model. Application developers should observe the behavior of the phone they are tapping.

Bit	Description
7	
6	Left square is "on" 
5	
4	Center square is "on" 
3	Music symbol 
2	Right square is "on" 

Phone Action Events

The following section highlights the observed variations noted with this particular PBX.

Digit Pressed

Both the 4020 and the 4035 phone models have keyboard functionality. When keyboard buttons are pressed, the EVT_DIGIT_PRESSED event is reported. The subreason field is used to pass over the exact character that was typed in ASCII format.

D-channel events per Phone Model

A complete list of the D-channel events observed when tapping the Alcatel is provided at the beginning of this chapter. AudioCodes has observed that the types of D-channel events reported may vary per phone model, installation or software version.

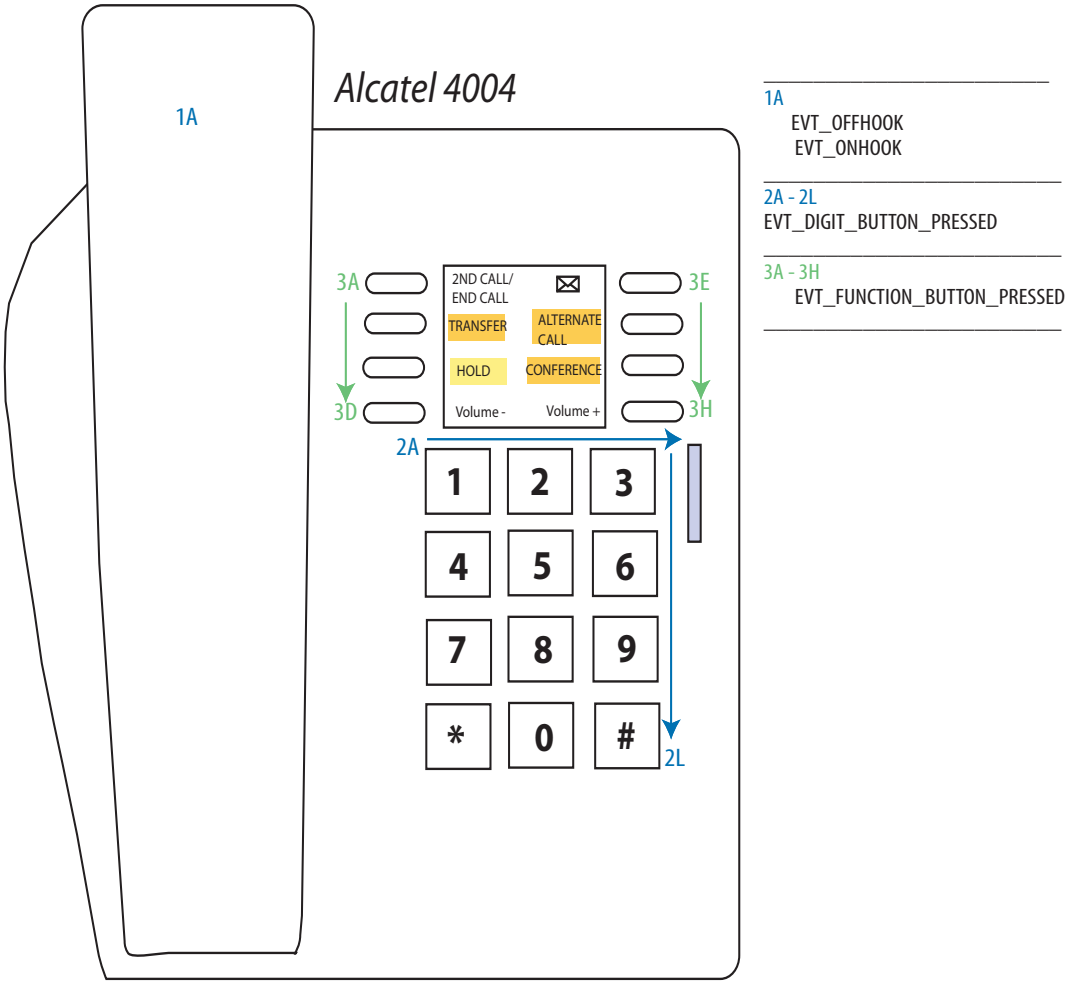
The following section can be used by an application developer to understand variations of D-channel events noted between phone models. This is not meant to be an exhaustive list, but rather an aide to application developers who are getting started.

NOTE: All data in this section was obtained while using the Alcatel 2400 and the Alcatel OmniPCX 4400 with the UA16 line card (software version is not documented). If another software version or line card is used, different D-channel patterns may be observed.

4004

Phone Map

The following events were observed when each phone button was used.



1A
EVT_OFFHOOK
EVT_ONHOOK

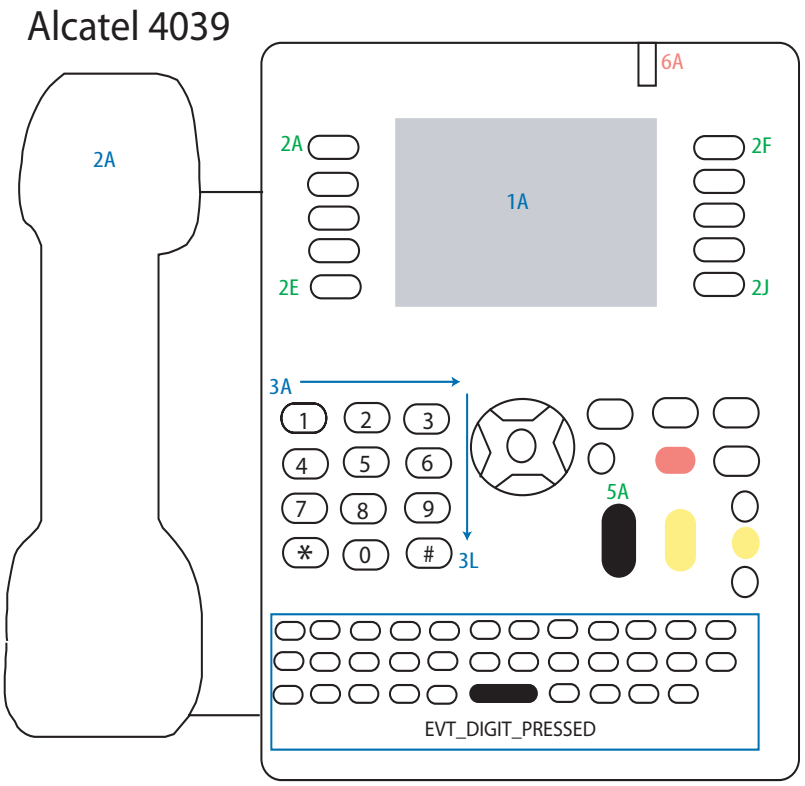
2A - 2L
EVT_DIGIT_BUTTON_PRESSED

3A - 3H
EVT_FUNCTION_BUTTON_PRESSED

4039

PHONE MAP

The following events were observed when each phone button was used.



1A
EVT_MESSAGE_CJHANGE
EVT_DISPLAY_CLOCK

2A
EVT_ONHOOK
EVT_OFFHOOK

3A-3L
EVT_DIGIT_BUTTON_PRESSED
The type of button pressed is passed to the user application in ASCII format

4A - 4J
EVT_FUNCTION/SOFT_BUTTON_PRESSED

5A EVT_RELEASE_BUTTON_PRESSED

6A EVT_FUNCTION_LIGHT_
ON, OFF, FLASHING

CALL SCENARIOS

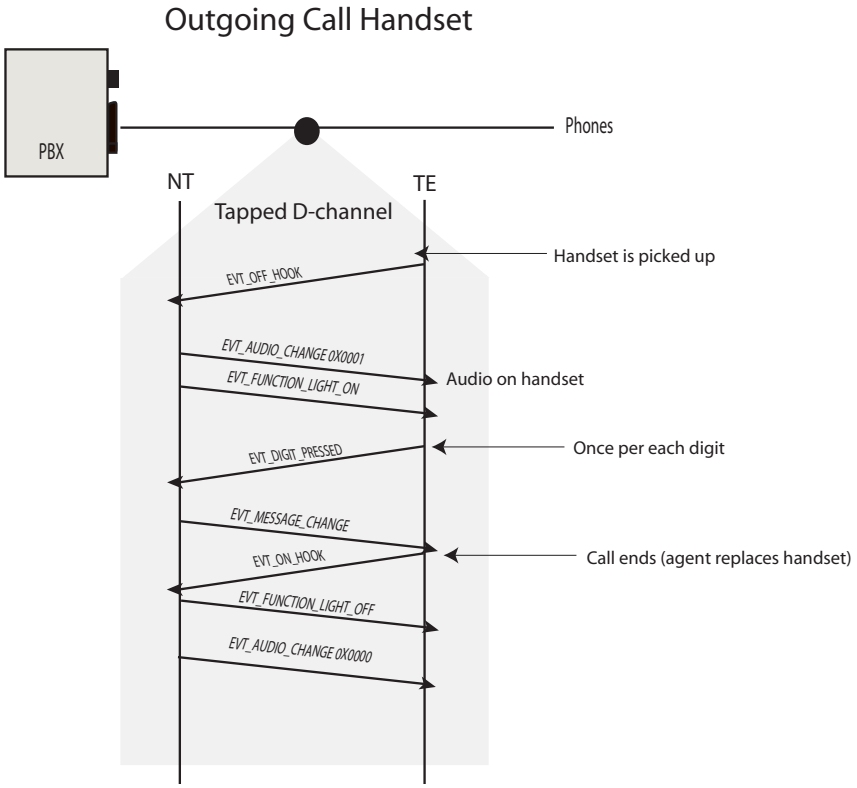
The following section shows what events were observed during a typical call scenario. These call scenarios were not generated by using a specific phone model and should be considered 'generic'. Application developers are encouraged to test with the phone model that will be tapped.

All calls were initiated when the phone is in an idle state:

- When applicable, the handset is ON_HOOK
- All lights are off
- LCD is clear or the clock is displayed.

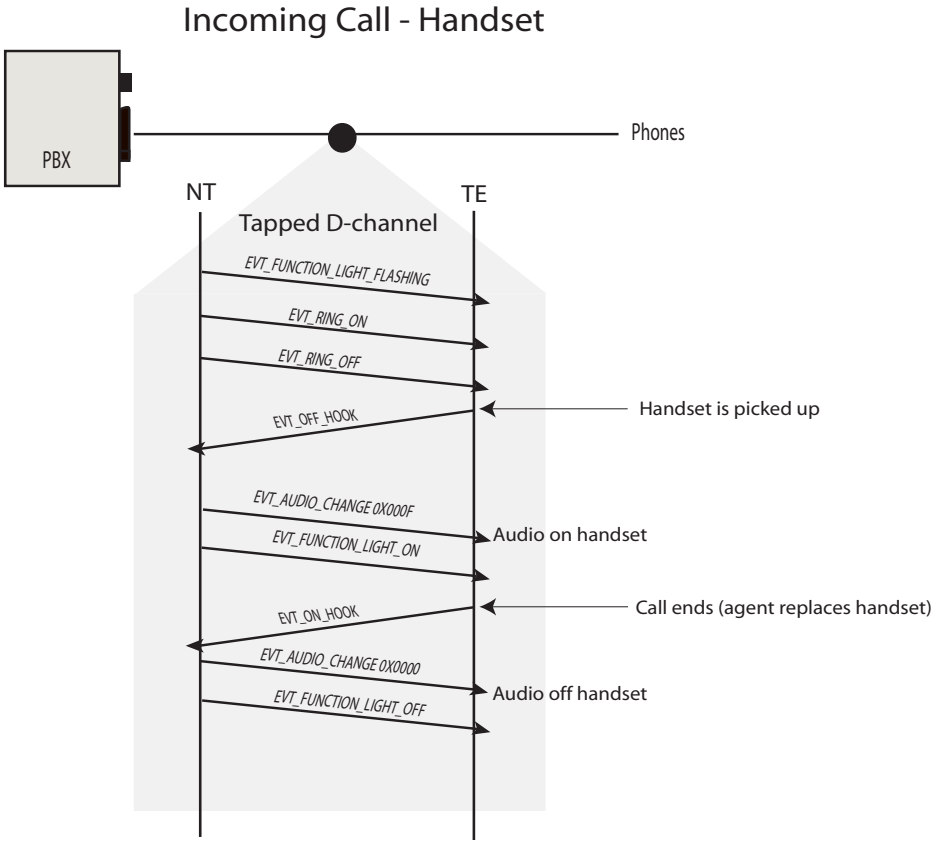
Outgoing Call - Handset

In this scenario the agent picks up the handset and dials a number. To end the call, the agent hangs up the phone.



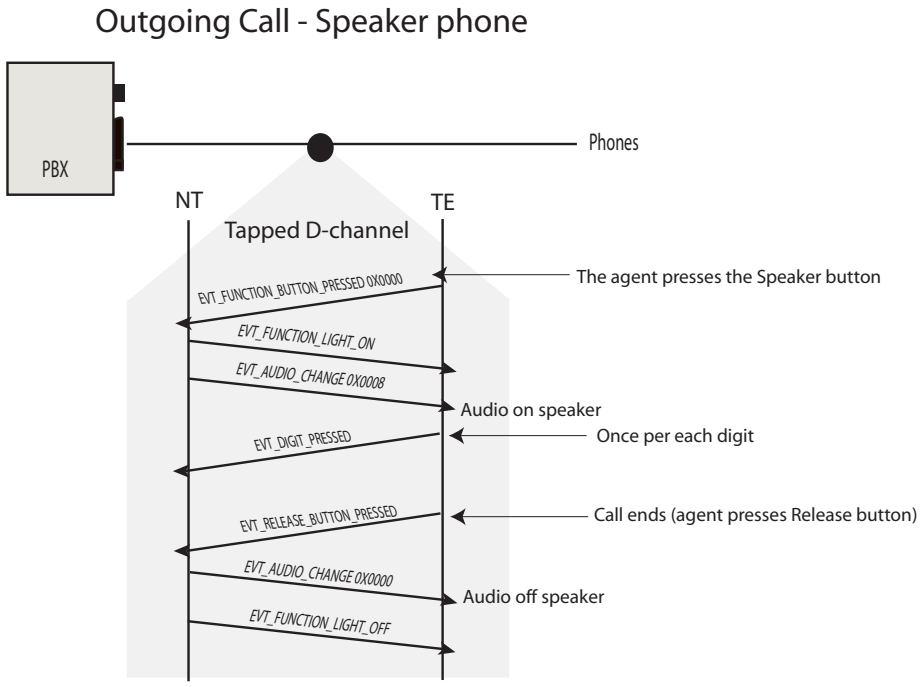
Incoming Call - Handset

An incoming call is received and the agent takes the call. The call is released when the agent replaces the handset.



Outgoing Call - Speaker Phone

The agent captures a line and dials the number. The call is terminated when the agent releases the line.



Chapter 5

Avaya Definity 2W

This chaper highlights the use of the SmartWORKS NGX when tapping the Avaya Definity 2W.

NOTE: All data in this section was obtained with the Avaya Definity 2W PBX with G3V7i.01.0.343.7 software and TN2181 or TN2224 line card. If another software version or line card is used, different D-channel patterns may be observed.

Phone Model Support

The following table shows the phone models that have been tested in a tapped environment. This information is updated frequently. For an updated copy refer to the Online Support system's Files and Documents section in the NGX_PBX_MATRIX folder.

Model	Vox		D-Channel	
8410	✓	T	✓	T
8434	✓	T		R
6402(display)	✓	T	✓	
6408(display)	✓	T	✓	T
6416(display)	✓	T	✓	
6424(display)	✓	T	✓	T
Call Master IV	✓	T	✓	S
Call Master V	✓	T	✓	T
Call Master VI	✓	T	✓	T

Status:
T - tested in house
S - supported based on product family (not tested)
R - tested by third party
N - not tested, it may work
W - tested, will not work

Installation and Configuration

Complete installation and wiring diagrams are available in the *SmartWORKS User's Guide*. The following section highlights important installation notes for this particular PBX.

Installation

This PBX is a two wire (2W) model. Follow the standard 2W installation procedures outlined in the *SmartWORKS User's Guide*.

Cable Lengths

The following are the maximum cable lengths (in feet) recommended by AudioCodes:

PBX to tap - 75'

tap to NGX - 300'

tap to phones - 2445'

CONFIGURATION

After the NGX card has been installed on the digital network, specific configuration steps are required to generate D-channel events.

CLOCK SETTINGS

AudioCodes recommends setting the NGX's clock source to the PBX. This avoids any synchronization problems. Use the ***API MTSetCTMasterClock()*** to set the clock source parameter to NET1. The default setting is to local - the board's clock (OSC).

PBX FIRMWARE

Using the Control Panel, select the **Board** tab. The *PBX Type* field must be set. Use the drop-down menu to select the name of the tapped PBX - Avaya Definity 2W. Click APPLY (this is required!). The firmware used for this PBX will be automatically installed the next time the board is restarted. Repeat this step for each NGX board and daughter card.

TDM ENCODING

Using the Control Panel, select the **Board** tab. The *TDM Encoding* field must be set. Generally speaking, µ-Law is used in North America and Japan, A-Law is used in Europe. This setting is also PBX dependant. It is highly recommended that you check PBX documentation and verify the setting used by the local network.

By default, the Avaya Definity 2W is set to µ-Law. This is a programmable value and AudioCodes recommends checking this setting prior to configuring the NGX.

If this setting is not correct, the recording may sound "metallic".

D-CHANNEL EVENTS

The NGX board does not generate D-channel events by default. This must be enabled. Using the Control Panel select the **Board** tab. The *D-channel* option must be enabled.

EVENT UPDATES

Many PBXs send duplicate commands or phone status reports over the line to the phones. As a result, the NGX decodes this information and generates extraneous events. This can be controlled by disabling the *Event Updates* option in the Control Panel (under Board tab). Once this is disabled, all duplicate events are filtered.

The Avaya Definity 2W sends phone status information every five minutes. If the Event Updates option is disabled in the Control Panel, these events are filtered by the NGX.

D-Channel Events

The following is a list of all D-channel events reported when tapping Avaya Definity 2W PBXs. All events have been grouped by event type.

Results vary depending on the configuration of the PBX in the field, along with the phone model used at the customer site. AudioCodes does not guarantee that all events are reported at each PBX site.

PBX Command Events

The following events are reported from commands passing from the PBX to the phones.

Call State Events

EVT_ABANDONED (indicates a far side hang up)

Signaling Events

No signaling events are decoded.

Audio Events

No audio events are decoded.

LED (Light) Events

EVT_FUNCTION_LIGHT_FASTFLASHING
EVT_FUNCTION_LIGHT_FLASHING
EVT_FUNCTION_LIGHT_OFF
EVT_FUNCTION_LIGHT_ON
EVT_FUNCTION_LIGHT_QUICKFLASH
EVT_FUNCTION_LIGHT_VERY_FASTFLASHING

Display (LCD) Events

EVT_DISPLAY_CLEAR
EVT_MESSAGE_CHANGE

Phone (Action) Commands

The following events are reported from data generated by the phone and passed to the PBX.

Hook State Events

EVT_OFF_HOOK
EVT_ON_HOOK

Button Depression Events

EVT_ANSWER_BUTTON_PRESSED
EVT_CONF_BUTTON_PRESSED
EVT_EXIT_BUTTON_PRESSED
EVT_FUNCTION_BUTTON_PRESSED
EVT_HOLD_BUTTON_PRESSED
EVT_MENU_BUTTON_PRESSED
EVT_NEXT_BUTTON_PRESSED
EVT_REDIAL_BUTTON_PRESSED
EVT_RELEASE_BUTTON_PRESSED
EVT_SOFT_BUTTON_PRESSED
EVT_TRANSFER_BUTTON_PRESSED

Avaya Definity 2W Behavior

Each PBX exhibits unique behaviors. This section shows how common line conditions are handled by the Avaya Definity 2W. This section is not meant to be an exhaustive list, but rather an overview of some of the behavior observed by AudioCodes.

NOTE: The following topics are explained in detail in the PBX Integration chapter of this book.

Dialed Numbers (DTMF) Detection

The NGX does not decode DTMF D-channel information for Avaya Definity 2W. To obtain DTMF, user applications must rely on the NGX DTMF detectors to detect in-band DTMF tones.

CallerID

On proprietary PBX networks, callerID is sometimes displayed on phone LCDs. This varies depending on the configuration of the PBX. As a result, AudioCodes cannot guarantee that CallerID is available when tapping the Avaya Definity 2W.

When the phone's LCD is updated by the PBX for any reason, the NGX decodes this information and the event EVT_MESSAGE_CHANGE is reported. The ptrBuffer field of the MT_EVENT structure points to the buffer and the datalength field is populated with the size of the data buffer pointed to by ptrBuffer, including a null character terminator at the end. **NOTE:** The callerID information must be parsed from the string.

Missed Actions

The Avaya Definity 2W does not pass audio events over the line, as a result these actions are not detected by the NGX. The EVT_AUDIO_CHANGE event is not generated.

CRC Error Checking

The Avaya Definity 2W relies on a 16-bit CRC check. Invalid messages are not decoded, and error message are logged in the system event viewer.

PBX Command Events

The following section highlights the observed variations noted with this particular PBX.

Signalling Events - Dial Tone

Whether the PBX or phone generates the tone is dependant on the system configuration when using the Avaya Definity 2W. It is up to the user to determine where the dial tone originated.

If the dial tone is generated by the PBX, the audio signal is detected by the NGX's Call Progress Monitoring (CPM) system and the event EVT_MON_DIAL is generated. CPM detectors are on the primary input only. If the tone is generated by the phone, these detectors do not pick up the signal.

For dial tone detection to occur, the signal generated must match the profile used by the Call Progress Monitoring system. Refer to the application note: *Understanding Signal Profiles*.

Dial tone is never reported with a D-channel event.

SIGNALLING EVENTS - RINGING TONES

The EVT_RING_ON and EVT_RING_OFF D-channel events are not available when using the Avaya Definity 2W. To alert the agent of an incoming phone, the PBX commands a light on the phone flash. The application developer must locate the light that is activated when an incoming call is present and rely on the corresponding EVT_FUNCTION_LIGHT_(on/flashing) event.

CALL STATE EVENTS - EVT_ABANDONED

The EVT_ABANDONED event indicates the far-end has disconnected from a connected call state. The Definity **only** provides this message when the far-end hangs up first. This is different from the definition used when monitoring call states on a BRI network: the EVT_CC_CALL_ABANDONED occurs when the calling party disconnects from the call attempt in the alerting state.

LCD DISPLAY EVENTS

Only EVT_MESSAGE_CHANGE and EVT_DISPLAY_CLEAR are reported with the Avaya Definity 2W .

The EVT_MESSAGE_CHANGE event is generated when the PBX passes a message over to the phone with the command to display it. These messages usually includes data such as agent ID, or caller ID. The data that the phone is commanded to display is held in a buffer and can be accessed by the user application. The ptrBuffer field of the MT_EVENT structure points to the buffer and the datalength field is populated with the size of the data buffer pointed to by ptrBuffer, including a null character terminator at the end. **NOTE:** The callerID and agent ID information must be parsed from the string.

LED LIGHT EVENTS

Many buttons on Avaya Definity 2W phones are associated with two lights. It is possible that both lights may be active at the same time - either both are turned ON or one is ON while the other is flashing. The NGX decodes each as a separate light, however the light number is the same for both. Users must rely on the light color represented in the subreason field for more information. Refer to illustrations of phone models for more information.

The following light behavior has been observed:

EVT_FUNCTION_LIGHT_OFF - idle

EVT_FUNCTION_LIGHT_ON - call is active or feature is active

EVT_FUNCTION_LIGHT_FLASHING - found during a call ringing

EVT_FUNCTION_LIGHT_FASTFLASHING - found during call on hold

EVT_FUNCTION_LIGHT_VERY_FASTFLASHING - found during a conference or transfer

EVT_FUNCTION_LIGHT_QUICKFLASH -found when pressing un-programmed function buttons.

Subreason field:

The light subreason field indicates the light number and color. Represented as a hex value the following holds true 0xRRRRCCNN where R = reserved, C = color, and N = light number. The following table represents each bit value of the subreason field:

RRRR	CC			NN
b31-b16	b15-b10	b9	b8	b7-b0
reserved	reserved	Red	Green	Light Number

PHONE (ACTION) EVENTS

The following section highlights the observed variations noted with this particular PBX.

EVT_SPEAKER_BUTTON

A handset call looks identical to a speaker call. Though the speaker key is available on most phones, the EVT_SPEAKER_BUTTON_PRESSED is not decoded.

D-channel Events per Phone Model

A complete list of the D-channel events observed when tapping the Avaya Definity 2W is provided at the beginning of this chapter. AudioCodes has observed that the types of D-channel events reported may vary per phone model, installation or software version.

The following section can be used by an application developer to understand the variations noted between phone models. This is not meant to be an exhaustive list, but rather an aide to application developers who are getting started.

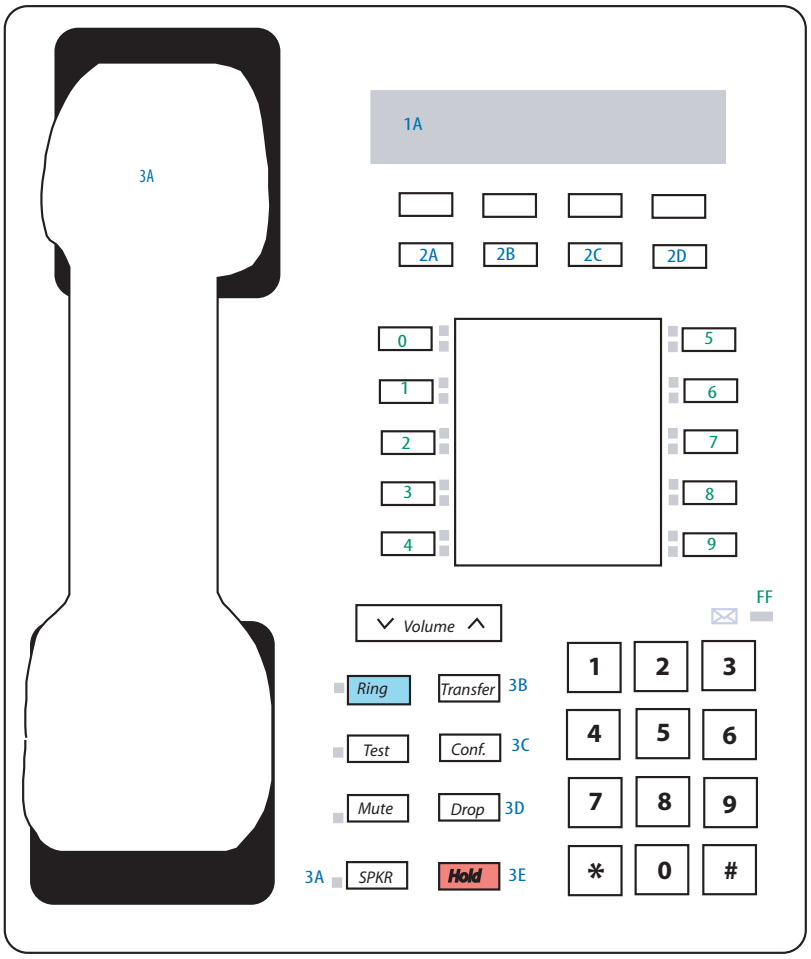
NOTE: All data in this section was obtained with the Avaya Definity 2W PBX with G3V7i.01.0.343.7 software and TN2181 or TN2224 line card. If another software version or line card is used, different D-channel data may be observed.

8410D

PHONE MAP

The following events were observed when each phone button was used.

Avaya 8410



1A
EVT_MESSAGE_CJHANGE
EVT_DISPLAY_CLEAR

2A EVT_MENU_BUTTON_PRESSED
2B EVT_EXIT_BUTTON_PRESSED
2C EVT_PREVIOUS_BUTTON_PRESSED
2D EVT_NEXT_BUTTON_PRESSED

3A EVT_OFFHOOK
EVT_ONHOOK
(Speaker button and headset)
3B EVT_TRANSFER_BUTTON_PRESSED
3C EVT_CONFERENCE_BUTTON_PRESSED
3D EVT_RELEASE_BUTTON_PRESSED
3E EVT_HOLD_BUTTON_PRESSED

0-9
EVT_FUNCTION_BUTTON_PRESSED
EVT_FUNCTION_LIGHT_FASTFLASHING
EVT_FUNCTION_LIGHT_FLASHING
EVT_FUNCTION_LIGHT_ON
EVT_FUNCTION_LIGHT_OFF
EVT_FUNCTION_LIGHT_QUICKFLASH
EVT_FUNCTION_LIGHT_VERY_FASTFLASHING
Subreason: 0x00000000
Bits 0-7 = Light Number
Bits 8 = Green
Bit 9 = Red

FF
EVT_FUNCTION_LIGHT_ON
EVT_FUNCTION_LIGHT_OFF
Subreason: 0x00000000
Bits 0-7 = Light Number(FF)
Bit 9 = Red

6408 D+(Display)

Refer to the section that explains the 6424D+ phone for information.

6416D+ (Display)

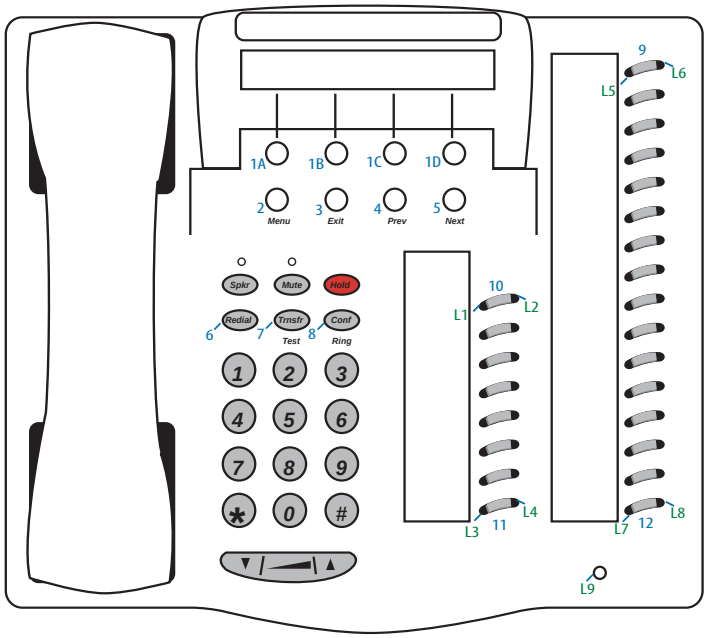
Refer to the section that explains the 6424D+ phone for information.

6424D+ (Display)

Phone Map

The following events were observed when each phone button was used.

Avaya 6424D+



EVT_SOFT_BUTTON_PRESSED
The subreason depends on weather the 1st, 2nd or 3rd menu options are displayed
Subreasons:
1A 0x0001, 0x0005, 0x0009
1B 0x0002, 0x0006,0x0010
1C 0x0003, 0x0007,00001
1D 0x0004, 0x0008,00012

2 EVT_MENU_BUTTON_PRESSED
3 EVT_EXIT_BUTTON_PRESSED
4 EVT_PREVIOUS_BUTTON_PRESSED
5 EVT_NEXT_BUTTON_PRESSED
6 EVT_REDIAL_BUTTON_PRESSED
7 EVT_TRANSFER_BUTTON_PRESSED
8 EVT_CONF_BUTTON_PRESSED
9 EVT_FUNCTION_BUTTON_PRESSED
Subreasons: 0x00000008
10 EVT_FUNCTION_BUTTON_PRESSED
Subreasons: 0x00000000
11 EVT_FUNCTION_BUTTON_PRESSED
Subreasons: 0x00000007
12 EVT_FUNCTION_BUTTON_PRESSED
Subreasons: 0x00000017

EVT___FUNCTION_LIGHT_FLASHING
EVT_FUNCTION_LIGHT_FAST_FLASHING
EVT_FUNCTION_LIGHT_QUICKFLASH
EVT_FUNCTION_LIGHT_VERY_FASTFLASHING

L1 Light No: 0
Color: Green
Subreason: 0x00000100

L2 Light No: 0
Color: Red
Subreason: 0x00000200

L3 Light No: 7
Color: Green
Subreason: 0x00000107

L4 Light No: 7
Color: Red
Subreason: 0x00000207

L5 Light No: 8
Color: Green
Subreason: 0x00000108

L6 Light No: 8
Color: Red
Subreason: 0x00000208

L7 Light No: 23
Color: Green
Subreason: 0x00000117

L8 Light No: 23
Color: Red
Subreason: 0x00000217

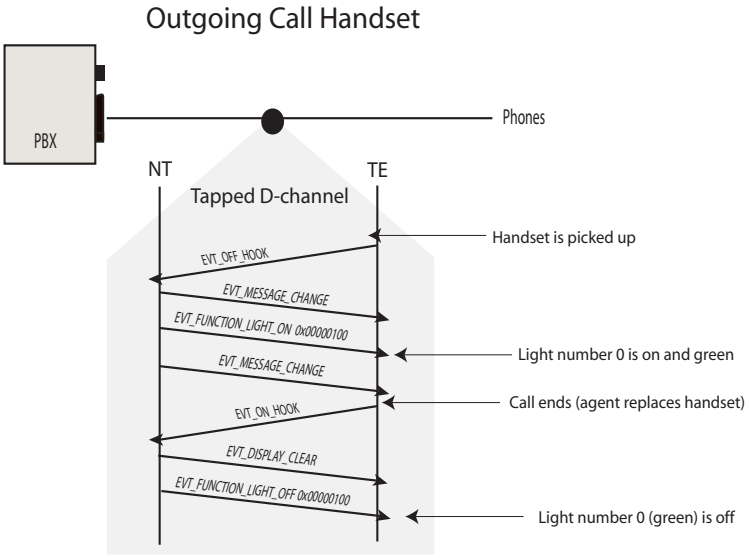
EVT_LIGHT_ON, EVT_LIGHT_OFF
L9 Light No: FF (Mail Indicator)
Color: RED
Subreason: 0x000002FF

CALL SCENARIOS

The following section shows what events were observed during a typical call scenario:

Outgoing Call - Handset

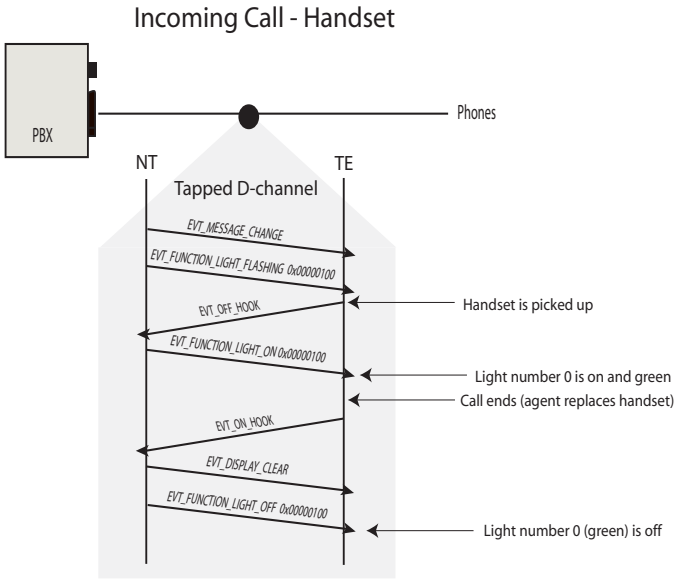
In this scenario the agent picks up the handset and dials a number. To end the call, the agent hangs up the phone.



The subreason field of the light events (EVT_FUNCTION_LIGHT_) is the light number and color.

Incoming Call - Handset

An incoming call is received and the agent takes the call. The call is released when the agent replaces the handset.

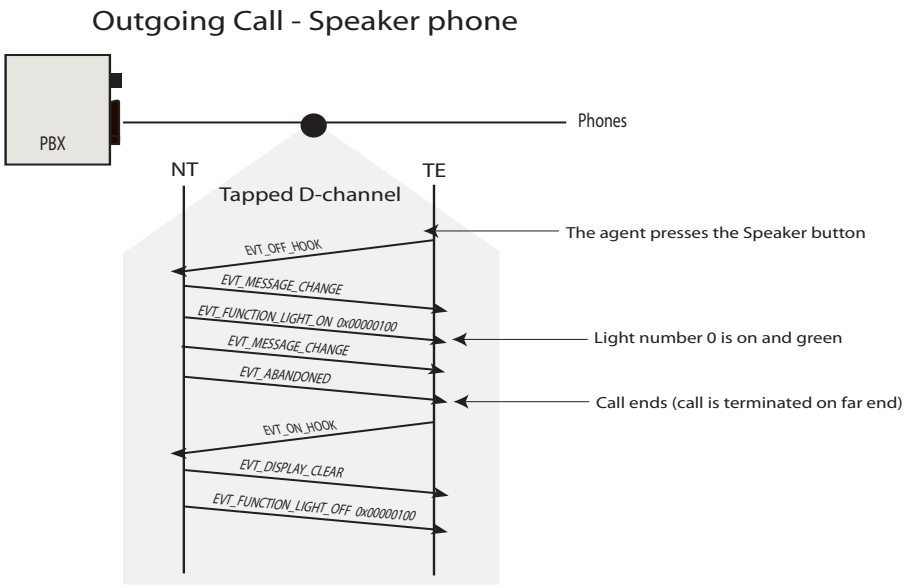


NOTES:
1. When EVT_MESSAGE_CHANGE is generated, the screen data is contained in a buffer. This typically contains caller ID, or agent ID.
2. The subreason field of the light events (EVT_FUNCTION_LIGHT_) is the light number and color

Outgoing Call - Speaker Phone

The EVT_SPEAKER_BUTTON is not supported. When this option is selected by the call agent EVT_OFF_HOOK/ON_HOOK is reported. As a result, handset and speaker calls look identical.

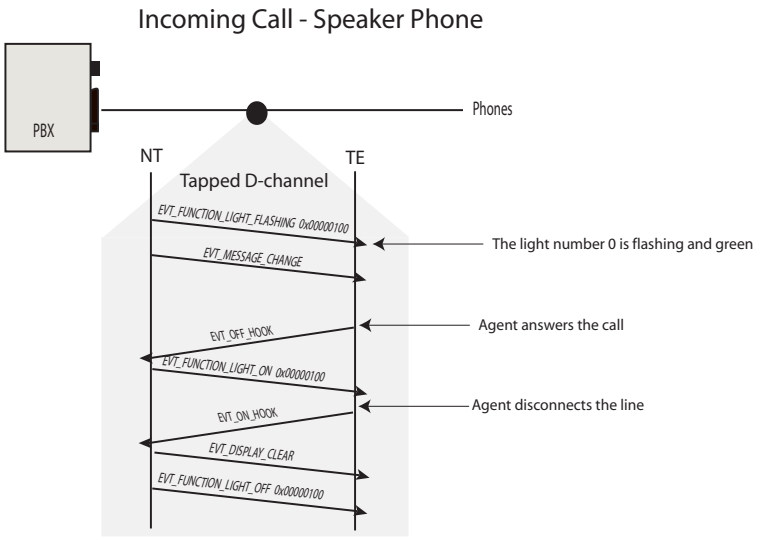
The agent captures a line and dials the number. The call is terminated on the far end.



The subreason field of the light events (EVT_FUNCTION_LIGHT_) is the light number.

Incoming Call - Speaker Phone

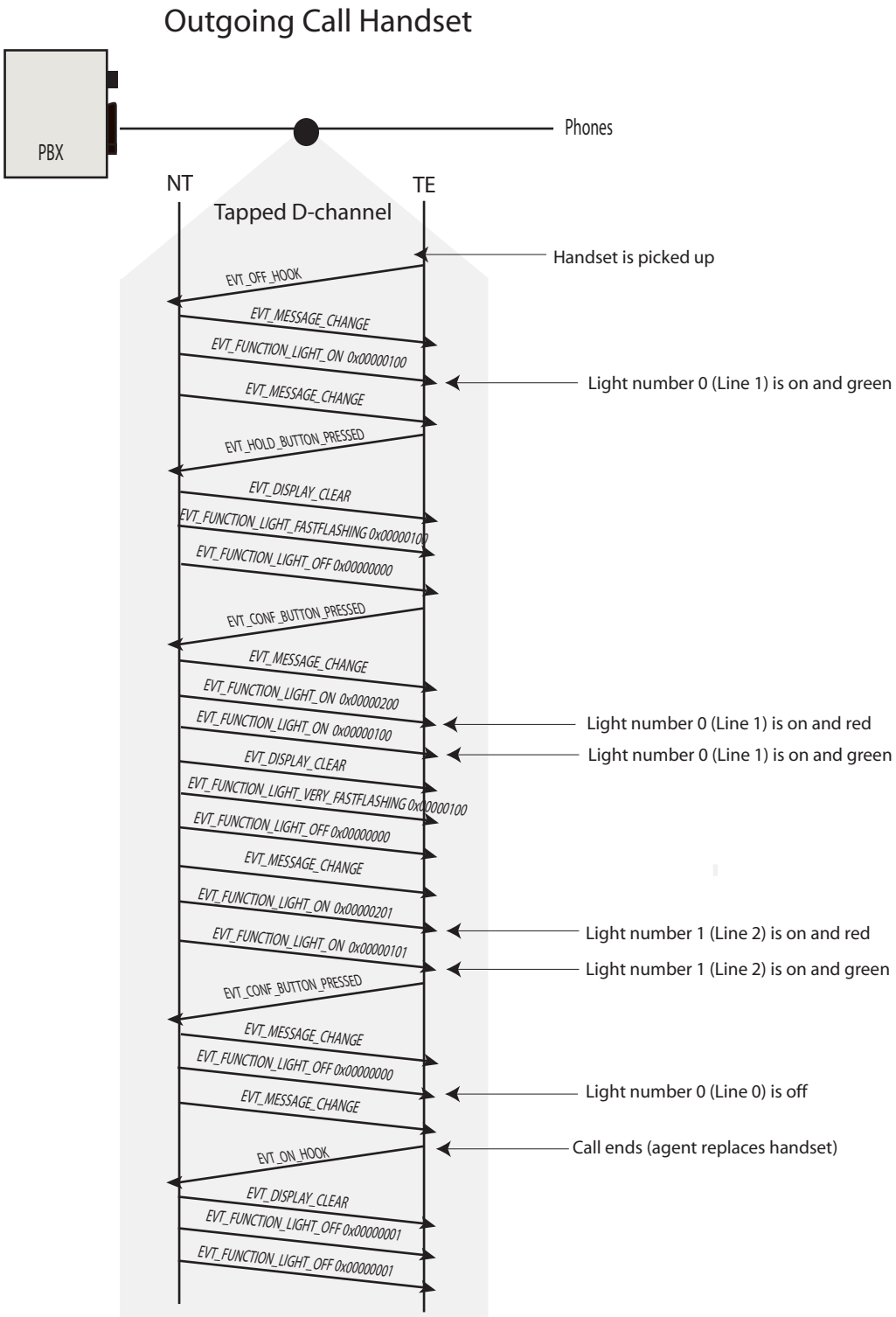
An incoming call is received and the agent accepts the call. The agent terminates the call.



NOTES:
1. When EVT_MESSAGE_CHANGE is generated, the screen data is contained in a buffer. This typically contains caller ID, or agent ID.
2. The subreason field of the light events (EVT_FUNCTION_LIGHT_) is the light number.

Outgoing Call - Conference Call

The agent initiates a three way conference call. The call is terminated when the agent replaces the handset. In this test call Lines 1 & 2 are used with light numbers 0 & 1 respectively.



The subreason field of the light events (EVT_FUNCTION_LIGHT_) is the light number and color.

CALL MASTER IV

This phone works with both Avaya Definity 2W and 4W PBXs.

CALL MASTER V

This section provides a detailed look when using this phone by providing common call scenarios, and noted behaviors.

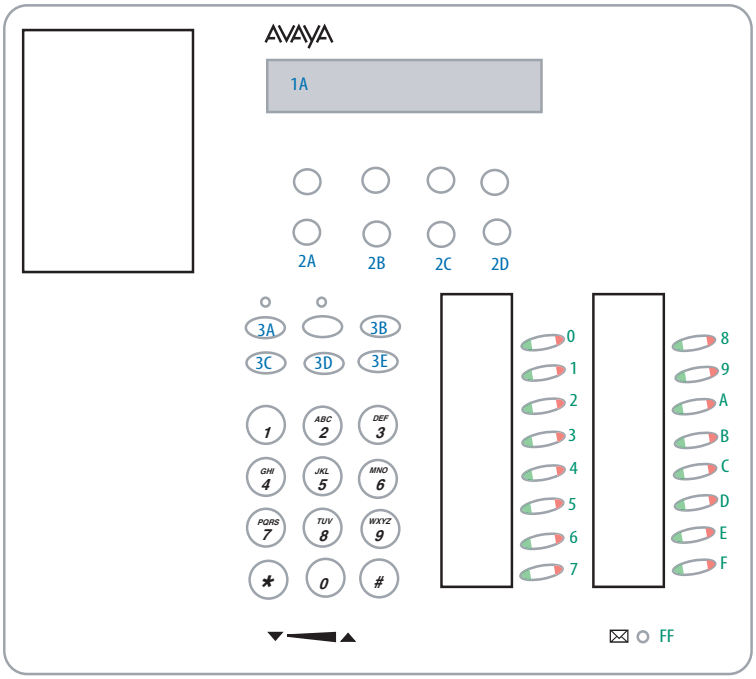
BEHAVIOR NOTED

- When a called party comes back into the conversation after placing the calling party on hold a "Display Timer" event is given

PHONE MAP

The following events were observed when each phone button was used.

Call Master V



1A
EVT_MESSAGE_CJHANGE
EVT_DISPLAY_CLEAR

2A EVT_MENU_BUTTON_PRESSED
2B EVT_EXIT_BUTTON_PRESSED
2C EVT_PREVIOUS_BUTTON_PRESSED
2D EVT_NEXT_BUTTON_PRESSED

3A EVT_OFFHOOK
EVT_ONHOOK
(Speaker button)
3B EVT_HOLD_BUTTON_PRESSED
3C EVT_REDIAL_BUTTON_PRESSED
3D EVT_TRANSFER_BUTTON_PRESSED
3E EVT_CONFERENCE_BUTTON_PRESSED

0-F
EVT_FUNCTION_BUTTON_PRESSED
EVT_FUNCTION_LIGHT_FASTFLASHING
EVT_FUNCTION_LIGHT_FLASHING
EVT_FUNCTION_LIGHT_ON
EVT_FUNCTION_LIGHT_OFF
EVT_FUNCTION_LIGHT_QUICKFLASH
EVT_FUNCTION_LIGHT_VERY_FASTFLASHING
Subreason: 0x00000000
Bits 0-7 = Light Number
Bits 8 = Green
Bit 9 = Red

FF
EVT_FUNCTION_LIGHT_ON
EVT_FUNCTION_LIGHT_OFF
Subreason: 0x00000000
Bits 0-7 = Light Number(FF)
Bit 9 = Red

CALL MASTER VI

This phone is supported but not documented.

Chapter 6

Avaya Definity 4W

This chapter highlights the use of the SmartWORKS NGX when tapping the Avaya Definity 4W.

NOTE: All data in this section was obtained with the Avaya Definity 4W PBX with G3V7i.01.0.343.7 software and TN270B, TN754, or TN754B line cards. If another software version or line card is used, different D-channel patterns may be observed.

Phone Model Support

The following table shows the phone models that have been tested in a tapped environment. This information is updated frequently. For an updated copy refer to the Online Support system's Files and Documents section in the NGX_PBX_MATRIX folder.

Model	Vox		D-Channel	
7406	✓	T	✓	T
8410	✓	T	✓	T
Call Master II	✓	T	✓	T

Status:
T - tested in house
S - supported based on product family (not tested)
R - tested by third party
N - not tested, it may work
W - tested, will not work

Installation and Configuration

Complete installation and wiring diagrams are available in the *SmartWORKS User's Guide*. The following section highlights important installation notes for this particular PBX.

Installation

This PBX is a four wire (4W) model. Follow the standard 4W installation procedures outlined in the *SmartWORKS User's Guide*.

Cable Lengths

The following are the maximum cable lengths (in feet) recommended by AudioCodes:

PBX to tap - 75'

tap to NGX - 300'

tap to phones - 2445'

Configuration

After the NGX card has been installed on the digital network, specific configuration steps are required to generate D-channel events.

CLOCK SETTINGS

AudioCodes recommends setting the NGX's clock source to the PBX. This avoids any synchronization problems. Use the ***API MTSetCTMasterClock()*** to set the clock source parameter to NET1. The default setting is to local - the board's clock (OSC).

PBX FIRMWARE

Using the Control Panel, select the **Board** tab. The *PBX Type* field must be set. Use the drop-down menu to select the name of the tapped PBX - *Avaya Definity 4W*. Click APPLY (this is required!). The firmware used for this PBX will be automatically installed the next time the board is restarted. Repeat this step for each NGX board and daughter card.

TDM ENCODING

Using the Control Panel, select the **Board** tab. The *TDM Encoding* field must be set. Generally speaking, µ-Law is used in North America and Japan, A-Law is used in Europe. This setting is also PBX dependant. It is highly recommended that you check PBX documentation to verify the setting required on your local network.

By default, the Avaya Definity 4W is set to µ-Law. This is a programmable value and AudioCodes recommends checking this setting prior to configuring the NGX.

If this setting is not correct, the recording may sound "metallic".

D-CHANNEL EVENTS

The NGX board does not generate D-channel events by default. This must be enabled. Using the Control Panel select the **Board** tab. The *D-channel* option must be enabled.

EVENT UPDATES

Many PBXs send duplicate commands or phone status reports over the line to the phones. As a result, the NGX decodes this information and generates extraneous events. This can be controlled by disabling the *Event Updates* option in the Control Panel (under Board tab). Once this is disabled, all duplicate events are filtered.

The Avaya Definity 4W sends phone status information every five minutes. If the Events Updates option is disabled in the Control Panel, these events are filtered by the NGX.

D-Channel Events

The following is a list of all D-channel events reported when tapping Avaya Definity 4W PBXs. All events have been grouped by event type.

Results vary depending on the configuration of the PBX in the field, along with the phone model used at the customer site. AudioCodes does not guarantee that all events are reported at each PBX site.

PBX COMMAND EVENTS

The following events are reported from commands passing from the PBX to the phones.

CALL STATE EVENTS

EVT_ABANDONED (indicates a far side hang up)

Signaling Events

No signaling events are reported for this PBX.

Audio Events

Audio events are not decoded.

LED (Light) Events

EVT_FUNCTION_LIGHT_FASTFLASHING
EVT_FUNCTION_LIGHT_FLASHING
EVT_FUNCTION_LIGHT_OFF
EVT_FUNCTION_LIGHT_ON
EVT_FUNCTION_LIGHT_QUICKFLASH
EVT_FUNCTION_LIGHT_VERY_FASTFLASHING

Display (LCD) Events

EVT_DISPLAY_CLEAR
EVT_MESSAGE_CHANGE

Phone (Action) Commands

The following events are reported from data generated by the phone and passed to the PBX.

Hook State Events

EVT_OFF_HOOK
EVT_ON_HOOK

Button Depression Events

EVT_ANSWER_BUTTON_PRESSED
EVT_CONF_BUTTON_PRESSED
EVT_EXIT_BUTTON_PRESSED
EVT_FUNCTION_BUTTON_PRESSED
EVT_HOLD_BUTTON_PRESSED
EVT_MENU_BUTTON_PRESSED
EVT_NEXT_BUTTON_PRESSED
EVT_REDIAL_BUTTON_PRESSED
EVT_RELEASE_BUTTON_PRESSED
EVT_SOFT_BUTTON_PRESSED
EVT_TRANSFER_BUTTON_PRESSED

Avaya Definity 4W Behavior

Each PBX exhibits unique behaviors. This section shows how common line conditions are handled by the Avaya Definity 4W. This section is not meant to be an exhaustive list, but rather an overview of some of the behavior observed by AudioCodes.

NOTE: The following topics are explained in detail in the PBX Integration chapter of this book.

Dialed Numbers (DTMF) Detection

The NGX does not decode DTMF D-channel information. To obtain DTMF, user applications must rely on the NGX DTMF detectors to isolate in-band DTMF tones.

Caller ID

On proprietary PBX networks, callerID is sometimes displayed on phone LCDs. This varies depending on the configuration of the PBX. As a result, AudioCodes cannot guarantee that CallerID is available when tapping the Avaya Definity 4W.

When the phone's LCD is updated by the PBX for any reason, the NGX decodes this information and the event EVT_MESSAGE_CHANGE is reported. The ptrBuffer field of the MT_EVENT structure points to the buffer that the "datalength" field is populated with the size of the data buffer pointed to by ptrBuffer, including a null character terminator at the end. **NOTE**: The callerID information must be parsed from the string.

Missed Actions

The Avaya Definity 4W does not pass audio events over the line, as a result these actions are not detected by the NGX. The EVT_AUDIO_CHANGE event is not generated.

CRC Error Checking

The Avaya Definity 4W relies on a 16-bit CRC check. Invalid messages are not decoded, and error message are logged in the system event viewer.

PBX Command Events

The following section highlights the observed variations noted with this particular PBX.

Signalling Events - Dial Tone

Whether the PBX or phone generates the tone is dependant on the system configuration when using the Avaya Definity 4W. It is up to the user to determine where the dial tone originated.

If the dial tone is generated by the PBX, the audio signal is detected by the NGX's Call Progress Monitoring (CPM) system and the event EVT_MON_DIAL is generated. CPM detectors are on the primary input only. If the tone is generated by the phone, these detectors do not pick up the signal.

For dial tone detection to occur, the signal generated must match the profile used by the Call Progress Monitoring system. Refer to the application note: *Understanding Signal Profiles*.

Signalling Events - Ringing Tones

The EVT_RING_ON and EVT_RING_OFF D-channel events are not available when using the Avaya Definity 4W.

Signalling Events - EVT_ABANDONED

For the EVT_ABANDONED with Definity, the event indicates the far-end has disconnected from a connected call state. The Definity **only** provides this message when the far-end hangs up first. This is different from the definition used when monitoring call states on a BRI network: the EVT_CC_CALL_ABANDONED occurs when the calling party disconnects from the call attempt in the alerting state.

Signalling Events - Incoming Call

The Avaya Definity 4W uses an alerting light on the phone to indicate an incoming call. The application developer must locate the light that is activated when an incoming call is present and rely on the corresponding EVT_FUNCTION_LIGHT_(on/flashing) event.

LCD Display Events

Only EVT_MESSAGE_CHANGE and EVT_DISPLAY_CLEAR are reported with the Avaya Definity 4W.

The EVT_MESSAGE_CHANGE event is generated when the PBX passes a message over to the phone with the command to display it. These messages usually includes data such as agent ID, or caller ID. The data that the phone is commanded to display is held in a buffer and can be accessed by the user application.

LED Light Events

Many buttons on Avaya Definity 4W phones are associated with two lights. It is possible that both lights may be active at the same time - either both are turned ON or one is ON while the other is flashing. The NGX decodes each as a separate light, however the light number is the same for both. Users must rely on the light color represented in the subreason field for more information. Refer to illustrations of phone models for more information.

The following light behavior has been observed:

EVT_FUNCTION_LIGHT_OFF - idle

EVT_FUNCTION_LIGHT_ON - call is active or feature is active

EVT_FUNCTION_LIGHT_FLASHING - found during a call ringing

EVT_FUNCTION_LIGHT_FASTFLASHING - found during call on hold

EVT_FUNCTION_LIGHT_VERY_FASTFLASHING - found during a conference or transfer

EVT_FUNCTION_LIGHT_QUICKFLASH -found when pressing un-programmed function buttons.

Subreason Field

The light subreason field indicates the light number and color. Represented as a hex value the following holds true 0xRRRRCCNN where R = reserved, C = color, and N = light number. The following table represents each bit value of the subreason field:

RRRR	CC			NN
b31-b16	b15-b10	b9	b8	b7-b0
reserved	reserved	Red	Green	Light Number

Phone Action Events

The following section highlights the observed variations noted with this particular PBX.

EVT_SPEAKER_BUTTON

A handset call looks identical to a speaker call. Though the speaker key is available on most phones, the EVT_SPEAKER_BUTTON_PRESSED is not decoded.

D-channel events per Phone Model

A complete list of the D-channel events observed when tapping the Avaya Definity 4W is provided at the beginning of this chapter. AudioCodes has observed that the types of D-channel events reported may vary per phone model, installation or software version.

The following section can be used by an application developer to understand the variations noted between phone models. This is not meant to be an exhaustive list, but rather an aide to application developers who are getting started.

NOTE: All data in this section was obtained with the Avaya Definity 4W PBX with G3V7i.01.0.343.7 software and TN270B, TN754, or TN754B line cards. If another software version or line card is used, different D-channel data may be observed.

7406

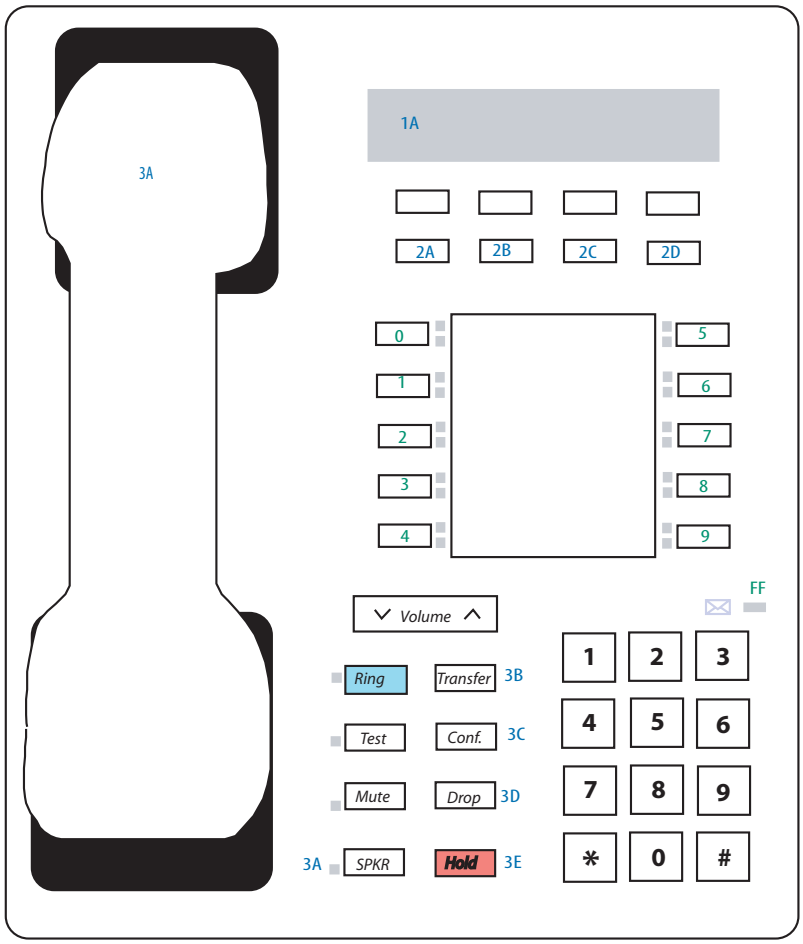
Tested, but not documented.

8410

PHONE MAP

The following events were observed when each phone button was used.

Avaya 8410



1A
EVT_MESSAGE_CJHANGE
EVT_DISPLAY_CLEAR

2A EVT_MENU_BUTTON_PRESSED
2B EVT_EXIT_BUTTON_PRESSED
2C EVT_PREVIOUS_BUTTON_PRESSED
2D EVT_NEXT_BUTTON_PRESSED

3A EVT_OFFHOOK
EVT_ONHOOK
(Speaker button and headset)
3B EVT_TRANSFER_BUTTON_PRESSED
3C EVT_CONFERENCE_BUTTON_PRESSED
3D EVT_RELEASE_BUTTON_PRESSED
3E EVT_HOLD_BUTTON_PRESSED

0-9
EVT_FUNCTION_BUTTON_PRESSED
EVT_FUNCTION_LIGHT_FASTFLASHING
EVT_FUNCTION_LIGHT_FLASHING
EVT_FUNCTION_LIGHT_ON
EVT_FUNCTION_LIGHT_OFF
EVT_FUNCTION_LIGHT_QUICKFLASH
EVT_FUNCTION_LIGHT_VERY_FASTFLASHING
Subreason: 0x00000000
Bits 0-7 = Light Number
Bits 8 = Green
Bit 9 = Red

FF
EVT_FUNCTION_LIGHT_ON
EVT_FUNCTION_LIGHT_OFF
Subreason: 0x00000000
Bits 0-7 = Light Number(FF)
Bit 9 = Red

CALL MASTER II

Tested, but not documented.

CALL MASTER IV

This phone works with both Avaya Definity 2W and 4W PBXs. It has not been tested with the SmartWORKS NGX while using a Definity 4W.

Chapter 7

Avaya INDeX

This chapter highlights the use of the SmartWORKS NGX when tapping the Avaya INDeX.

NOTE: The line card and software version used while testing this PBX in the AudioCodes lab has not been documented.

Phone Model Support

The following table shows the phone models that have been tested in a tapped environment. This information is updated frequently. For an updated copy refer to the Online Support system's Files and Documents section in the NGX_PBX_MATRIX folder.

Model	Vox		D-Channel	
DT3	✓	T	✓	T
2030	✓	T	✓	T
2050	✓	T	✓	T

Status:
T - tested in house -
S - supported based on product family (not tested)
R - tested by third party
N - not tested, it may work
W - tested, will not work

Installation and Configuration

Complete installation and wiring diagrams are available in the *SmartWORKS User's Guide*. The following section highlights important installation notes for this particular PBX.

Installation

This PBX is a two wire (2W) model. However, a four wire (4W) installation is required when using the AudioCodes RTX box. Refer to the *SmartWORKS User's Guide* for more wiring information.

Cable Lengths

The following are the maximum cable lengths (in feet) recommended by AudioCodes:

PBX to tap - 30*

tap to NGX - 30*

tap to phones - 1500*

* These values have not be tested, but are a fair estimate. AudioCodes customers have deployed the NGX with this PBX, however, we cannot provide exact cable length specifications. Recommended cable lengths are published in the PBX Support folder on the Online Support system. This document is continuously updated as more systems are tested.

Using an AudioCodes RTS

The Avaya Index uses a full duplex data transmission design where PBX and phone data is transmitted simultaneously. The NGX is designed to decode a half-duplex system, where phone and PBX data is transmitted in turns (also referred to as the

ping pong method). To decode this PBX an AudioCodes RTS (Resistive Tap Splitter) must be installed on the line. The RTS component splits the full-duplex signal into two half duplex signals. As a result, this 2-wire PBX is interpreted as a 4-wire PBX by the NGX.

CONFIGURATION

After the NGX card has been installed on the digital network, specific configuration steps are required to generate D-channel events.

CLOCK SETTINGS

AudioCodes recommends setting the NGX's clock source to the PBX. This avoids any synchronization problems. Use the ***API MTSetCTMasterClock()*** to set the clock source parameter to NET1.

PBX FIRMWARE

Using the Control Panel, select the **Board** tab. The *PBX Type* field must be set. Use the drop-down menu to select the name of the tapped PBX - *Avaya INDeX*. Click APPLY (this is required!). The firmware used for this PBX will be automatically installed the next time the board is restarted. Repeat this step for each NGX board and daughter card.

TDM ENCODING

Using the Control Panel, select the **Board** tab. The *TDM Encoding* field must be set. Generally speaking, µ-Law is used in North America and Japan, A-Law is used in Europe. This setting is also PBX dependant. It is highly recommended that you check PBX documentation to verify the setting required on your local network.

By default, the Avaya INDeX is set to A-Law. This is a programmable value and AudioCodes recommends checking this setting prior to configuring the NGX.

If this setting is not correct, the recording may sound "metallic".

D-CHANNEL EVENTS

The NGX board does not generate D-channel events by default. This must be enabled. Using the Control Panel select the **Board** tab. The *D-channel* option must be enabled.

EVENT UPDATES

Many PBXs send duplicate commands or phone status reports over the line to the phones. As a result, the NGX decodes this information and generates extraneous events. This can be controlled by disabling the *Event Updates* option in the Control Panel (under Board tab). Once this is disabling, all duplicate events are filtered.

The Avaya INDeX does not send phone status information. In this scenario, it is not necessary to disable the Event Updates option in the Control Panel.

D-Channel Events

The following is a list of all D-channel events reported when tapping Avaya INDeX PBXs. All events have been grouped by event type.

Results vary depending on the configuration of the PBX in the field, along with the phone model used at the customer site. AudioCodes does not guarantee that all events are reported at each PBX site.

PBX Command Events

The following events are reported from commands passing from the PBX to the phones.

Signalling Events

EVT_RING_OFF
EVT_RING_ON

Call State Events

No call state events are reported for this PBX.

Audio Events

EVT_AUDIO_CHANGE

LED (Light) Events

EVT_DIVERT_LIGHT_FASTFLASHING
EVT_DIVERT_LIGHT_FLASHING
EVT_DIVERT_LIGHT_OFF
EVT_DIVERT_LIGHT_ON
EVT_DND_LIGHT_FASTFLASHING
EVT_DND_LIGHT_FLASHING
EVT_DND_LIGHT_OFF
EVT_DND_LIGHT_ON
EVT_FUNCTION_LIGHT_FASTFLASHING
EVT_FUNCTION_LIGHT_FLASHING
EVT_FUNCTION_LIGHT_OFF
EVT_FUNCTION_LIGHT_ON
EVT_GROUP_LIGHT_FASTFLASHING
EVT_GROUP_LIGHT_FLASHING
EVT_GROUP_LIGHT_OFF
EVT_GROUP_LIGHT_ON
EVT_SCROLL_LIGHT_FASTFLASHING
EVT_SCROLL_LIGHT_FLASHING
EVT_SCROLL_LIGHT_OFF
EVT_SCROLL_LIGHT_ON
EVT_SOFT_LIGHT_FASTFLASHING
EVT_SOFT_LIGHT_FLASHING
EVT_SOFT_LIGHT_OFF
EVT_SOFT_LIGHT_ON
EVT_SPEAKER_LIGHT_FASTFLASHING
EVT_SPEAKER_LIGHT_FLASHING
EVT_SPEAKER_LIGHT_OFF
EVT_SPEAKER_LIGHT_ON

Display (LCD) Events

EVT_DISPLAY_CLOCK
EVT_DISPALY_TIMER
EVT_MESSAGE_CHANGE

Phone (Action) Commands

The following events are reported from data generated by the phone and passed to the PBX.

Hook State Events

EVT_OFFHOOK
EVT_ONHOOK

Button Depression Events

EVT_ANSWER_BUTTON_PRESSED
EVT_DIGIT_PRESSED
EVT_DIVERT_BUTTON_PRESSED
EVT_DND_BUTTON_PRESSED (do no disturb)
EVT_FUNCTION_BUTTON_PRESSED
EVT_GROUP_BUTTON_PRESSED
EVT_HOLD_BUTTON_PRESSED
EVT_MUTE_BUTTON_PRESSED
EVT_PROGRAM_BUTTON_PRESSED
EVT_REDIAL_BUTTON_PRESSED
EVT_SCROLL_BUTTON_PRESSED
EVT_SOFT_BUTTON_PRESSED
EVT_SPEAKER_BUTTON_PRESSED
EVT_SPEEDDIAL_BUTTON_PRESSED

Avaya INDeX Behavior

Each PBX exhibits unique behaviors. This section shows how common line conditions are handled by the Avaya INDeX. This section is not meant to be an exhaustive list of PBX behavior, but rather an aid to developers who are getting started.

NOTE: The following topics are explained in detail in the PBX Integration chapter of this book.

Dialed Numbers (DTMF) Detection

When tapping the Avaya INDeX the DTMF is passed in-band. This signal is detected with the NGX's DTMF detectors and reported as EVT_DIGIT. When the phone buttons are used to dial numbers, this information is decoded from the D-channel and reported as EVT_DIGIT_PRESSED. In both scenarios, the exact digit (in ASCII format) is presented to the user application in the subreason field of the MT_EVENT structure.

CallerID

On proprietary PBX networks, callerID is sometimes displayed on phone LCDs. This varies depending on the configuration of the PBX. As a result, AudioCodes cannot guarantee that CallerID is available when tapping the Avaya INDeX.

When the phone's LCD is updated by the PBX for any reason, the NGX decodes this information and the event EVT_MESSAGE_CHANGE is reported. The ptrBuffer field of the MT_EVENT structure points to the buffer that the "datalength" field is populated with the size of the data buffer pointed to by ptrBuffer, including a null character terminator at the end. **NOTE:** The callerID information must be parsed from the string.

CRC Error Checking

The Avaya INDeX relies on a 16-bit CRC check. Invalid messages are not decoded, and error message are logged in the system event viewer.

PBX Command Events

The following section highlights the observed variations noted with this particular PBX.

Signal Events - Call Progress Tones

When using the Avaya INDeX the phone generates all tones, nothing is played through the B-channels. Therefore call progress tones such as dial tone, ringing tone, or busy tone are not detected by the NGX CPM feature. The PBX commands the phone to generate tones. PBX documentation describes the following types of tones:

dial tone - dial number for call
broken dial tone - played on divert or no call alert
repeated tone - busy number
triple tone - call diverting to external number
continuous tone - number called is unobtainable, set to no calls, or call is barred

The NGX only decodes commands that pertain to a ring tone and reports EVT_RING_ON/OFF.

Signalling Events - Ring Events

The PBX orders the phone to generate a ring tone. As a result, the D-channel event EVT_RING_ON is reported. Once the call agent answers the phone the event EVT_RING_OFF is reported. The user application must rely on the timestamp between these two events to determine how long the phone has been ringing on the agent side.

Ring Cadence

The following has been noted with the 2030 and 2050 phone models.

The subreason field indicates the tone, frequency and incremental steps that should be associated with this ring command - 0xTTSSVV. Where 0xTT = ringer tone frequency, 0xVV = volume, 0xSS = incremental steps for volume on ringer. This can be used to determine the type of ring tone generated by the phone.

The phone documentation specifies the following ring types available for the 2030 & 2050:

repeated single ring - incoming calls
repeated double ring - external calls
repeated triple ring - system or personal calls
single tone - page call

LCD Display Events

The NGX reports changes to the LCD with the following events: EVT_MESSAGE_CHANGE, EVT_DISPLAY_TIMER, EVT_DISPLAY_CLOCK.

When a called party comes back into the conversation after placing the calling party on hold a EVT_DISPLAY_TIMER event is given. This has only been noted on the DT3 and 2030 phone models.

AUDIO EVENTS - EVT_AUDIO_CHANGE

The PBX controls external devices such as speakers, microphones, or headsets. These commands are decoded and reported as EVT_AUDIO_CHANGE. The subreason field is used to identify the state of the device. Refer to the PBX Integration chapter of this book for information about the subreason field.

LED LIGHT EVENTS

The Avaya INDeX has a group of programmable function buttons which are referred to as DSS keys. AudioCodes uses the term Function Buttons. Each function button is associated with one LED which can be either green or red. The following information describes the events associated with the light's cadence (flashing) pattern.

Red BLF Lamp Signals: (Calls to/from other extensions)

EVT_FUNCTION_LIGHT_FLASHING - Slow Flash: Call on line parked at another extension

EVT_FUNCTION_LIGHT_FASTFLASHING - Fast Flash: Extension is ringing

EVT_FUNCTION_LIGHT_ON - Solid: Extension or line disconnected, extension or line is busy

NOTE: Avaya INDeX documentation notes a Long On/Long Off cadence pattern. This was not observed by Ai_Logix.

Green BLF Lamp Signals(Calls to agent extension)

EVT_FUNCTION_LIGHT_FLASHING - Slow Flash: Call on line parked at agent's extension

EVT_FUNCTION_LIGHT_FASTFLASHING - Fast Flash: External call ringing on your extension

EVT_FUNCTION_LIGHT_ON - Solid: Call connected or held at agent's extension

Alternate Green/Red Lamp Signal (Pilot Number)

This action is not decoded, no event is generated. Slow Flash: Caller is ringing pilot number.

Subreason field:

When the EVT_FUNCTION_LIGHT_ON/OFF/FLASHING/FASTFLASHING is reported the subreason field indicates the light number and color. Represented as a hex value the following holds true 0xRRRRCCNN where R = reserved, C = color, and N = light number. The following table represents each bit value of the subreason field:

RRRR	CC			NN
b31-b16	b15-b10	b9	b8	b7-b0
reserved	reserved	Red	Green	Light Number

EVT_SOFT_LIGHT_

Events EVT_SOFT_LIGHT_ON, EVT_SOFT_LIGHT_OFF, EVT_SOFT_LIGHT_FLASHING, and EVT_SOFT_LIGHT_FASTFLASHING represent lights which appear in the phone's LCD. Each light is associated with a particular soft key (soft button) located around the phone's LCD. The subreason field indicates the light number. There are four soft key on the Avaya INDeX phones. The following information explains the position of the soft key and the number associated with it.

Soft Key "Left up" = 0x00
Soft Key "Right up" = 0x01
Soft Key "Left down"= 0x02
Soft Key "Right down" = 0x03

Phone Action Events

The following section highlights the observed variations noted with this particular PBX.

Button Depression Events - EVT_SOFT_BUTTON_PRESSED

Four unique subreason's are generated with the EVT_SOFT_BUTTON_PRESSED event. This has been noted on only the DT3 and 2030 phone models. The subreason field indicates the light number. There are four soft key on the Avaya INDeX phones. The following information explains the position of the soft key and the number associated with it.

Soft Key "Left up" = 0x00
Soft Key "Right up" = 0x01
Soft Key "Left down"= 0x02
Soft Key "Right down" = 0x03

D-channel Events per Phone Model

A complete list of the D-channel events observed when tapping the Avaya INDeX is provided at the beginning of this chapter. AudioCodes has observed that the types of D-channel events reported may vary per phone model, installation or software version.

The following section can be used by an application developer to understand the variations noted between phone models. This is not meant to be an exhaustive list, but rather an aide to application developers who are getting started.

NOTE: The line card and software version used while testing this PBX in the AudioCodes lab has not been documented.

DT3

This phone is supported, but not documented.

2030

This phone is supported but not documented.

Chapter 8

Nortel Meridian 1

This chapter highlights the use of the SmartWORKS NGX when tapping the Nortel Meridian 1.

NOTE: All data in this section was obtained with the Nortel Meridian 1 PBX with option 11C software and line card NT8D02GA, release 07. If another software version or line card is used, different D-channel patterns may be observed.

Phone Model Support

The following table shows the phone models that have been tested in a tapped environment. This information is updated frequently. For an updated copy refer to the Online Support system's Files and Documents section in the NGX_PBX_MATRIX folder.:

Model	Vox		D-Channel	
2006	✓	T	✓	T
2008	✓	T	✓	T
2216	✓	T	✓	T
2250	✓	T		
2616	✓	T	✓	T
2317	✓	T		
3901	✓	T	✓	T
3902	✓	T	✓	T
3903	✓	T	✓	T
3904	✓	T	✓	T
3905	✓	T	✓	T

Status:
T - tested in house
S - supported based on product family (not tested)
R - tested by third party
N - not tested, it may work
W - tested, will not work

Installation and Configuration

Complete installation and wiring diagrams are available in the *SmartWORKS User's Guide*. The following section highlights important installation notes for this particular PBX.

Installation

This PBX is a two wire (2W) model. Follow the standard 2W installation procedures outlined in the *SmartWORKS User's Guide*.

Cable Lengths

The following are the maximum cable lengths recommended by AudioCodes:

PBX to tap - 360'

tap to NGX - 100'

tap to phones - 1545'

CONFIGURATION

After the NGX card has been installed on the digital network, specific configuration steps are required to generate D-channel events.

CLOCK SETTINGS

AudioCodes recommends setting the NGX's clock source to the PBX. This avoids any synchronization problems. Use the ***API MTSetCTMasterClock()*** to set the clock source parameter to NET1. The default setting is to local - the board's clock (OSC).

PBX FIRMWARE

Using the Control Panel, select the **Board** tab. The *PBX Type* field must be set. Use the drop-down menu to select the name of the tapped PBX - *Nortel Meridian 1*. Click APPLY (this is required!). The firmware used for this PBX will be automatically installed the next time the board is restarted. Repeat this step for each NGX board and daughter card.

TDM ENCODING

Using the Control Panel, select the **Board** tab. The *TDM Encoding* field must be set. Generally speaking, µ-Law is used in North America and Japan, A-Law is used in Europe. This setting is also PBX dependant. It is highly recommended that you check PBX documentation to verify the setting required on your local network.

By default, the Nortel Meridian 1 is set to µ-Law. This is a programmable value and AudioCodes recommends checking this setting prior to configuring the NGX.

If this setting is not correct, the recording may sound "metallic".

D-CHANNEL EVENTS

The NGX board does not generate D-channel events by default. This must be enabled. Using the Control Panel select the **Board** tab. The *D-channel* option must be enabled.

EVENT UPDATES

Many PBXs send duplicate commands or phone status reports over the line to the phones. As a result, the NGX decodes this information and generates extraneous events. This can be controlled by disabling the *Event Updates* option in the Control Panel (under Board tab). Once this is disabled, all duplicate events are filtered.

The Nortel Meridian 1sends phone status information. If the Event Updates option is disabled in the Control Panel, these events are ignored.

D-Channel Events

The following is a list of all D-channel events reported when tapping Nortel Meridian 1 PBXs. All events have been grouped by event type.

Results vary depending on the configuration of the PBX in the field, along with the phone model used at the customer site. AudioCodes does not guarantee that all events are reported at each PBX site.

PBX Command Events

The following events are reported from commands passing from the PBX to the phones.

Signalling Events

EVT_RING_OFF
EVT_RING_ON

Audio Events

EVT_AUDIO_CHANGE

Call State Events (All of these events are Meridian specific)

No call state events are reported.

Phone State Events

NOTE: These types of events are Meridian specific and are used to report a change in the state of the phone.

EVT_CFWD
EVT_CFWD_CANCELED
EVT_AUTO_ANSWER
EVT_AUTO_ANSWER_CANCELED
EVT_SET_BUSY
EVT_SET_BUSY_CANCELED

LED (Light) Events

EVT_FUNCTION_LIGHT_FASTFLASHING
EVT_FUNCTION_LIGHT_FLASHING
EVT_FUNCTION_LIGHT_OFF
EVT_FUNCTION_LIGHT_ON

Display (LCD) Events

EVT_DISPLAY_CLEAR
EVT_DISPLAY_CLOCK
EVT_DISPLAY_MESSAGE
EVT_DISPLAY_TIMER
EVT_MESSAGE_CHANGE

Phone (Action) Commands

The following events are reported from data generated by the phone and passed to the PBX.

Hook State Events

EVT_OFFHOOK
EVT_ONHOOK

Button Depression Events

EVT_DIGIT_PRESSED
EVT_DIGIT_RELEASED
EVT_FUNCTION_BUTTON_PRESSED

EVT_FUNCTION_BUTTON_RELEASED
EVT_HOLD_BUTTON_PRESSED
EVT_HOLD_BUTTON_RELEASED
EVT_RELEASE_BUTTON_PRESSED
EVT_SPEAKER_BUTTON_PRESSED
EVT_SPEAKER_BUTTON_RELEASED

Nortel Meridian 1 Behavior

Each PBX exhibits unique behaviors. This section shows how common line conditions are handled by the Nortel Meridian 1. This section is not meant to be an exhaustive list, but rather an overview of some of the behavior observed by AudioCodes.

NOTE: The following topics are explained in detail in the PBX Integration chapter of this book.

Dialed Numbers (DTMF) Detection

When tapping the Nortel Meridian 1 the DTMF is not passed in-band. Digits can be detected when local phone buttons are used to dial numbers. This information is decoded from the D-channel and reported as EVT_DIGIT_PRESSED. The exact digit (in ASCII format) is presented to the user application in the subreason field of the MT_EVENT structure.

CallerID

On proprietary PBX networks, callerID is sometimes displayed on phone LCDs. This varies depending on the configuration of the PBX. As a result, AudioCodes cannot guarantee that CallerID is available when tapping the Nortel Meridian 1.

When the phone's LCD is updated by the PBX for any reason, the NGX decodes this information and the event EVT_MESSAGE_CHANGE is reported. The ptrBuffer field of the MT_EVENT structure points to the buffer and the "datalength" field is populated with the size of the data buffer pointed to by ptrBuffer, including a null character terminator at the end. **NOTE**: The callerID information must be parsed from the string.

Missed Actions

The Nortel Meridian 1 turns the audio state on and doesn't set the state to idle between calls on the 2216 and 3905 phones. When these phones are plugged in, the PBX sends the Audio ON command (EVT_AUDIO_CHANGE). The audio remains on for these phones, whether a call is in progress or not. This is known as the "hot mic" scenario. As a result, audio change events can not be used to determine call recording. The following lists other missed actions with the 2216 and 3905 phones:

- No Audio Off events are generated when a call is terminated
- No ring events are generated with incoming calls
- when an agent mutes the phone, this action is not passed from the phone to the PBX. No event is reported.

The 3905 phone does not have a handset. As a result no ON_HOOK, OFF_HOOK conditions can be reported.

CRC Error Checking

The Nortel Meridian 1 does not have a CRC check for validating D-channel packets. As a result, when line conditions are poor invalid data can be captured by the NGX and reported to the user application.

CRC errors usually indicate poor line conditions or improper wiring. It is important to monitor line conditions by using the error checking APIs: ***MTGetFramerStatus()*** and ***MTGetNGXFramerStatistic()***.

PBX Command Events

The following section highlights the observed variations noted with this particular PBX.

Signalling Events - Dial Tone

The Nortel Meridian 1 generates the dial tone and passes it in-band over the line to the phone. As a result, the Call Progress Monitoring feature is able to detect the signal and the event EVT_MON_DIAL is reported. The signal profiles configured by default on each channel are based on North American standards. The profile parameters of the dial tone must be modified to match the cadence and frequency values of the signal produced by the Nortel Meridian 1. Refer to the application notes: *Understanding Signal Profiles* for more information.

Signalling Events - Ring Tones

Nortel Meridian 1, sends one EVT_RING_ON and EVT_RING_OFF each time the phone rings. The user application must count the total number of EVT_RING_ON events to determine total ring count.

Phone State Event - Call Forward

At the time of this writing, the Nortel Meridian is the only PBX that generates an event for Call Forwarding (EVT_CFWD and EVT_CFWD_CANCELED). When the agent presses the call forward button to activate this feature, the EVT_CFWD event is reported. A buffer is provided that contains the phone number that the call is forwarded to. When the feature is deactivated by the call agent the event EVT_CFWD_CANCELED is reported.

Audio Events - EVT_AUDIO_CHANGE

The Nortel Meridian 1 turns the audio state on and doesn't set the state to idle between calls on the 2216 and 3905 phones. When these phones are plugged in, the PBX sends the Audio ON command (EVT_AUDIO_CHANGE). The audio remains on for these phones, whether a call is in progress or not. This is known as the “hot mic” scenario. As a result, audio change events can not be used to determine call recording.

LCD Display Events

When decoding Nortel Meridian 1 D-channel, the following LCD events are reported: EVT_MESSAGE_CHANGE, EVT_DISPLAY_CLCOK, EVT_DISPLAY_TIMER, EVT_DISPLAY_CLEAR and EVT_DISPLAY_MESSAGE.

When EVT_DISPLAY_MESSAGE is generated the subreason field of the MT_EVENT structure identifies the message ID. The following table lists the messages with corresponding message ID that have been observed when testing the PBX in the AudioCodes lab. This is not an exhaustive list. Not every message is supported on each phone.

Table 2: Observed Messages with Message ID

Subreason/Message ID	Message
0x0002	"Release and Try Again" (Reorder)

TABLE 2: OBSERVED MESSAGES WITH MESSAGE ID

0x0003	Call Forward Setting "Press CFWD or Enter New #"
0x0004	Call Forward Change Number "CFWD" "Enter Digits, Press CFWD"
0x0005	"CALL TRANSFER " "ENTER DIGITS "
0x0006	" " "TRANSFER WHEN READY"
0x0007	"CONFERENCE PARTY" "ENTER DIGITS"
0x0008	" " "CONFERENCE WHEN READY"
0x0009	"PROGRAMMING AUTODIAL" "ENTER NUMBER"
0x000A	" " "PRESS AUTIODIAL TO SAVE"
0x0010	"DESTINATION BUSY" "ACTIVATE RING AGAIN?"
0x0028	(this message is unknown, but the ID was captured at a customer site)
0x0071	"PROGRAMMING STORED#" "ENTER NUMBER"
0x0072	" " "PRESS STORED# TO SAVE"
0x008F	"Charge is not Allowed" " "
0x0092	"Press OK to park call on" " "

LED LIGHT EVENTS

The following light behavior has been observed when using the Nortel Meridian 1

EVT_FUNCTION_LIGHT_FASTFLASHING - the light flashes about two times per second. This has been observed when a call is on hold

EVT_FUNCTION_LIGHT_FLASHING - the light flashed about one time per second. This has been observed when the phone is ringing

EVT_FUNCTION_LIGHT_ON - a call is in progress, or when a feature is active

EVT_FUNCTION_LIGHT_OFF - phone is idle

Subreason field:

The light subreason field indicates the light number and color. Represented as a hex value the following holds true 0xRRRRCCNN where R = reserved, C = color, and N = light number. The phones that integrate with the Nortel Meridian 1 do not change colors. Therefore the color bits are not used. The following illustrations shows the bits values set in the subreason field when decoding the Nortel Meridian 1:

RRRR	CC	NN
b31-b16	b15-b8	b7-b0
reserved	reserved	Light Number

Phone Action Events

The following section highlights the observed variations noted with this particular PBX.

Speaker Button

Variations have been noticed when using the 2616 phone. When the function button that is identified by subreason 15 is used, two outcomes are possible. Normally this button yields the EVT_FUNCTION_BUTTON_PRESSED event. The PBX can be configured to use this button as a speaker phone button. When the class of service HFA is selected, then this button yields the EVT_SPEAKER_BUTTON_PRESSED/RELEASED events.

D-channel events per Phone Model

A complete list of the D-channel events observed when tapping the Nortel Meridian 1 is provided at the beginning of this chapter. AudioCodes has observed that the types of D-channel events reported may vary per phone model, installation or software version.

The following section can be used by an application developer to understand variations of D-channel events noted between phone models. This is not meant to be an exhaustive list, but rather an aide to application developers who are getting started.

NOTE: All data in this section was obtained while using the Nortel Meridian 1 PBX with option 11C software and line card NT8D02GA, release 07.

M2006

No information is available.

M2008

No information is available.

M2216

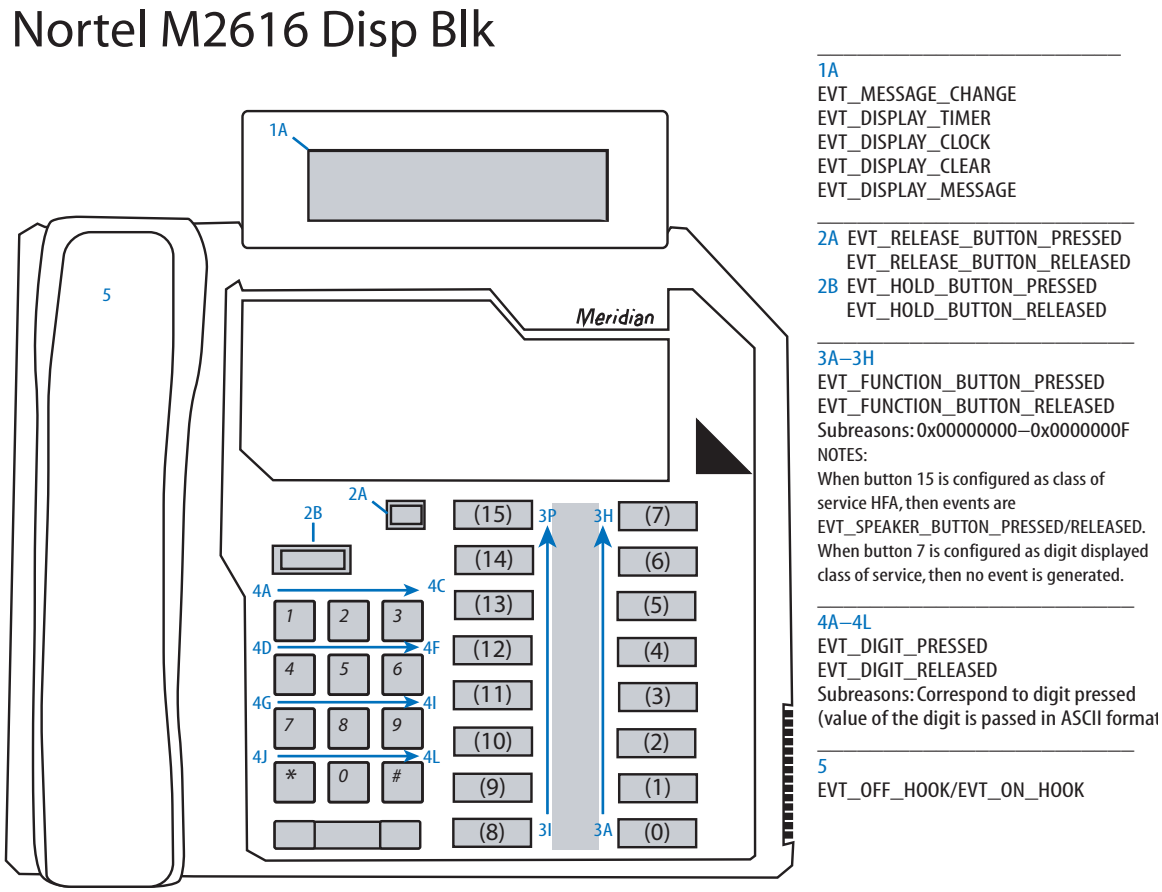
This phone does not have a handset, therefore EVT_OFF_HOOK and EVT_ON_HOOK events are not generated. This phone exhibits similar behavior to the 3905 phone model. Refer to the call scenarios provided in this document.

When this phone is plugged in, the PBX sends the Audio ON command (EVT_AUDIO_CHANGE). The audio remains on for this phone, whether a call is in progress or not. This is known as the “hot mic” scenario. As a result, audio change events can not be used to determine call recording.

M2616

Phone Map

The following events were observed when each phone button was used.



M3901

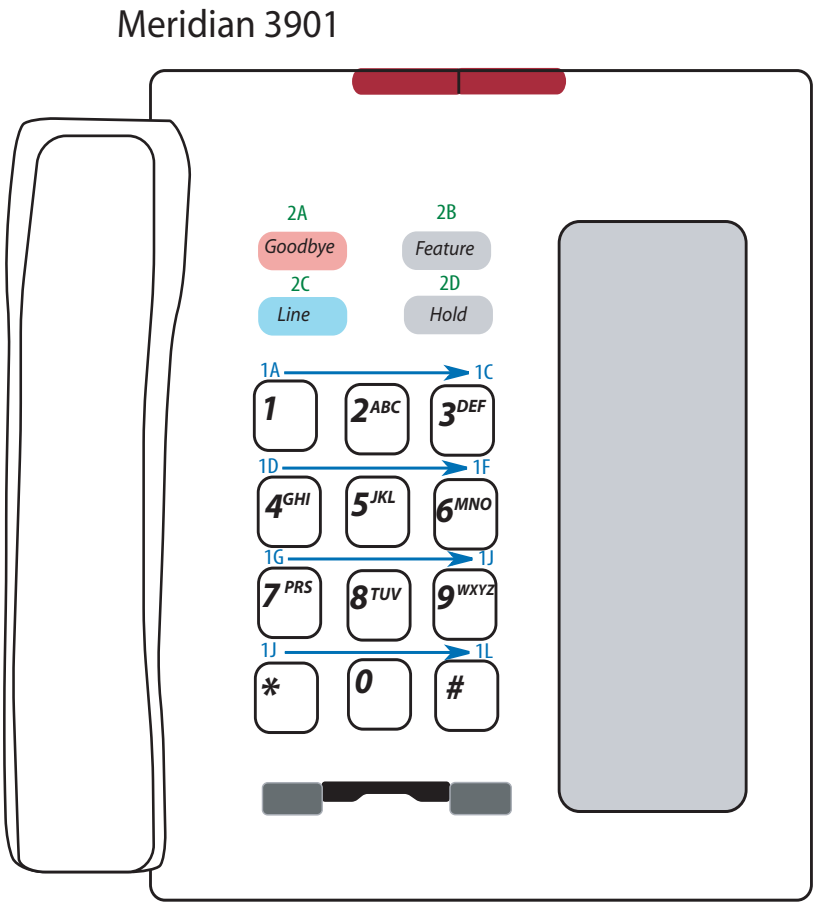
This section provides a detailed look when using this phone by providing a phone map, common call scenarios, and noted behaviors.

Noted Phone Behavior

- the line button can only be used to seize a line, it is not used to end a call
- the goodbye button is used to end a call

PHONE MAP

The following events were observed when each phone button was used.



1A–1L
EVT_DIGIT_PRESSED
EVT_DIGIT_RELEASED
Subreasons: 0x00000000–0x0000000C

2A EVT_RELEASE_BUTTON_PRESSED
EVT_RELEASE_BUTTON_RELEASED
2B This button acts as a shift key
2C EVT_FUNCTION_BUTTON_PRESSED
EVT_FUNCTION_BUTTON_RELEASED
2D EVT_HOLD_BUTTON_PRESSED
EVT_HOLD_BUTTON_RELEASED

EVT_OFF_HOOK and EVT_ONHOOK events are observed when the handset is used.

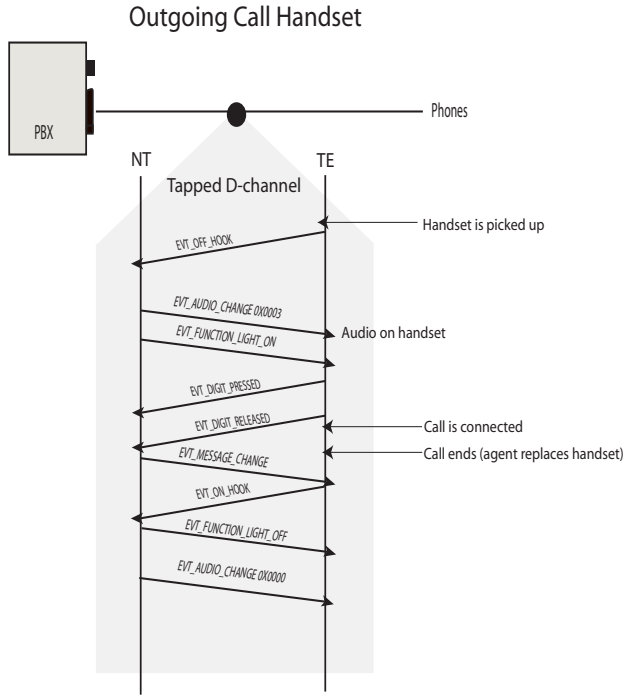
CALL SCENARIOS

The following section shows what events were observed during a typical call scenario. All calls were initiated when the phone is in an idle state:

- When applicable, the handset is ON_HOOK
- All lights are off
- LCD is clear or the clock is displayed.

Outgoing Call - Handset

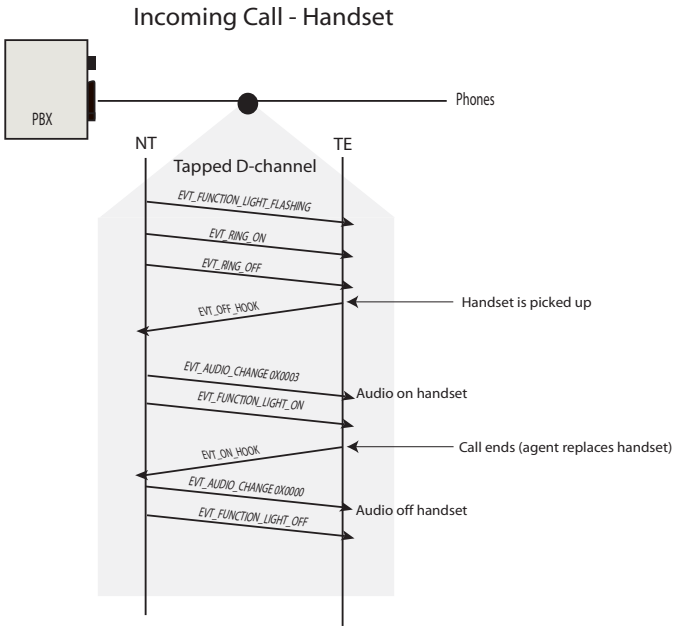
In this scenario the agent picks up the handset and dials a number. To end the call, the agent hangs up the phone.



The subreason field of the light events (EVT_FUNCTION_LIGHT_) is the light number.

Incoming Call - Handset

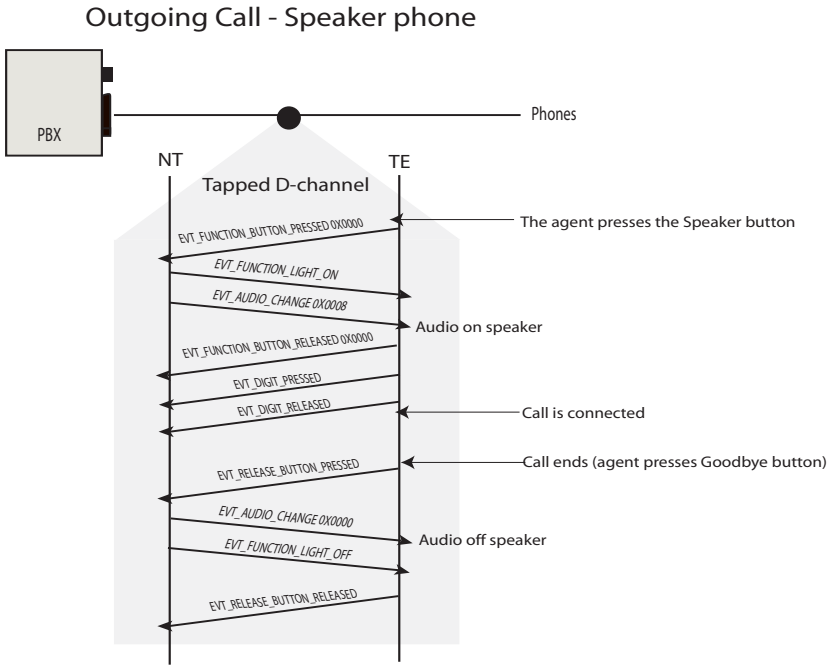
An incoming call is received and the agent takes the call. The call is released when the agent replaces the handset.



The subreason field of the light events (EVT_FUNCTION_LIGHT_) is the light number.

Outgoing Call - Speaker Phone

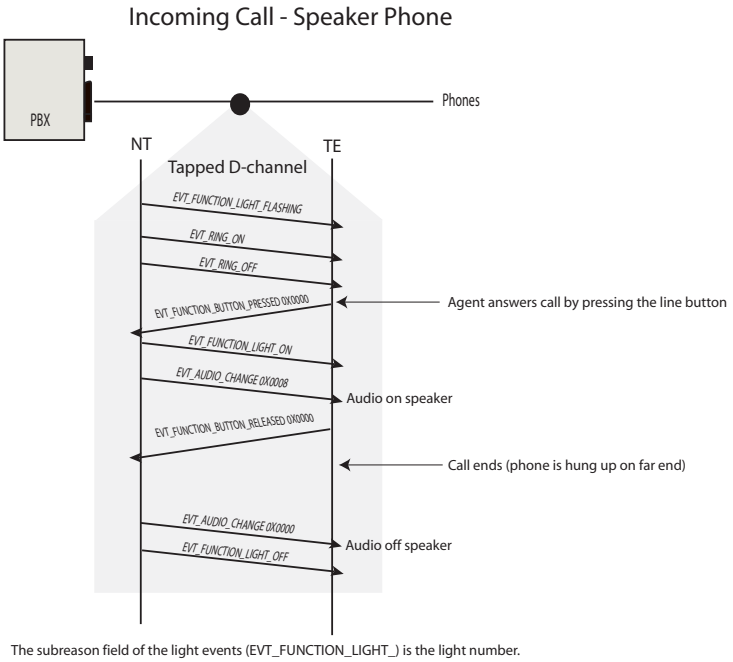
The agent captures a line and dials the number. The call is terminated when the agent releases the line.



The subreason field of the light events (EVT_FUNCTION_LIGHT_) is the light number.

Incoming Call - Speaker Phone

An incoming call is received and the agent accepts the call. In this scenario, the call is terminated when hung up on the far end.



M3902

No information is available at this time.

M3903

This section provides a detailed look when using this phone by providing common call scenarios, and noted behaviors.

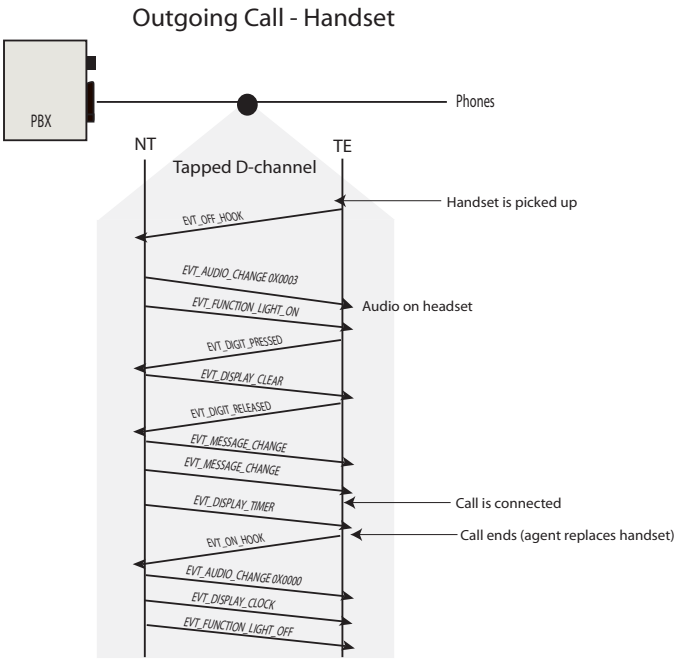
CALL SCENARIOS

The following section shows what events were observed during a typical call scenario. All calls were initiated when the phone is in an idle state:

- When applicable, the handset is ON_HOOK
- All lights are off
- LCD is clear or the clock is displayed

Outgoing Call - Handset

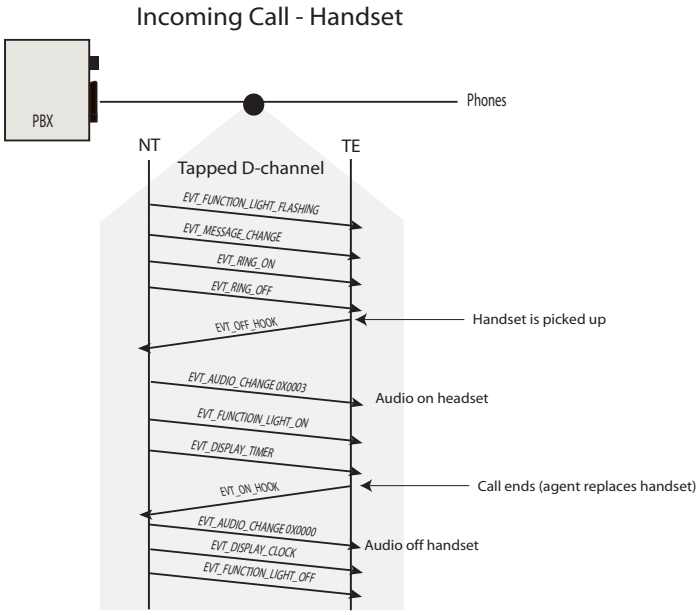
In this scenario the agent picks up the handset and dials a number. To end the call the agent hangs up the phone.



The subreason field of the light events (EVT_FUNCTION_LIGHT_) is the light number.

Incoming Call - Handset

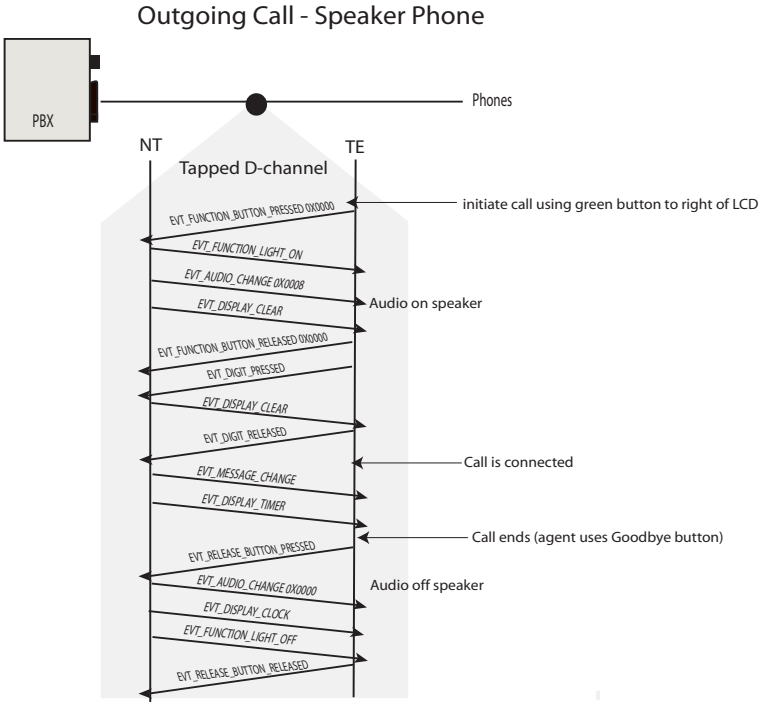
An incoming call is received and the agent takes the call. The call is released when the agent replaces the handset.



The subreason field of the light events (EVT_FUNCTION_LIGHT_) is the light number.

Outgoing Call - Speaker Phone

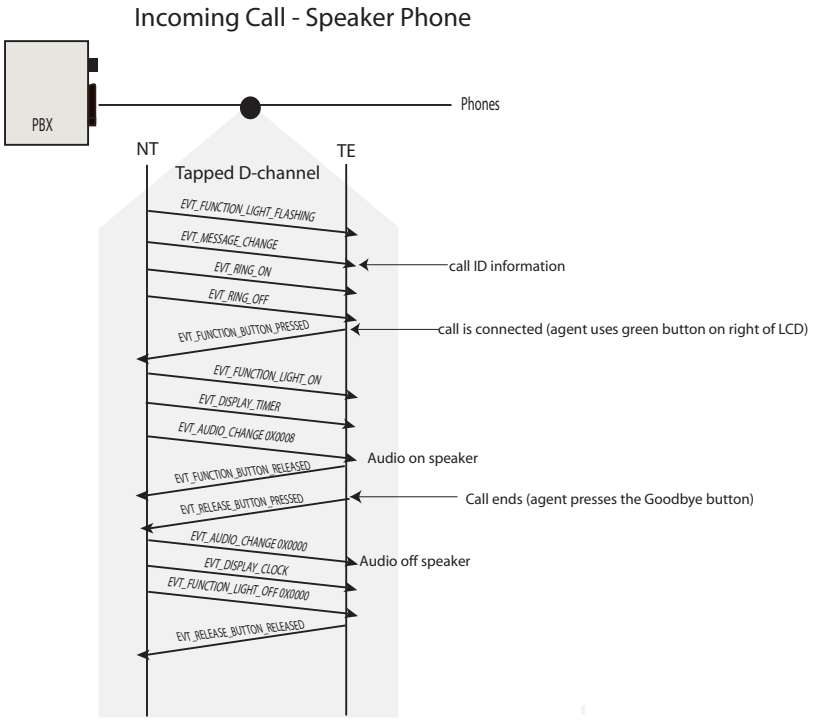
The agent captures a line and dials the number. The call is terminated when the agent releases the line.



The subreason field of the light events (EVT_FUNCTION_LIGHT_) is the light number.

Incoming Call - Speaker Phone

An incoming call is received and the agent accepts the call. The call is terminated when the agent releases the line.



The subreason field of the light events (EVT_FUNCTION_LIGHT_) is the light number.

M3904

No information is available at this time.

M3905

This section provides a detailed look when using this phone by providing a phone map, common call scenarios, and noted behaviors.

NOTED BEHAVIOR

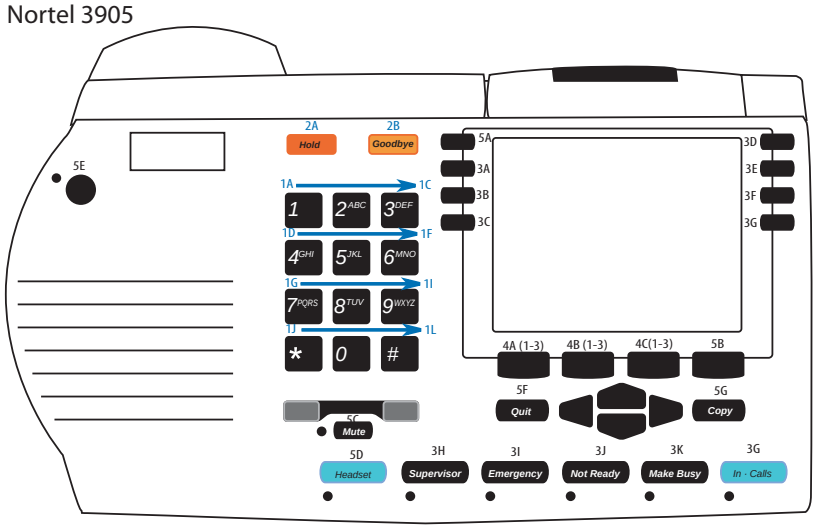
This phone does not have a handset, therefore EVT_OFF_HOOK and EVT_ON_HOOK events are not generated. This phone exhibits similar behavior to the 2216 phone model.

When this phone is plugged in, the PBX sends the Audio ON command (EVT_AUDIO_CHANGE). The audio remains on for this phone, whether a call is in progress or not. This is known as the “hot mic” scenario. As a result, audio change events can not be used to determine call recording.

PHONE MAP

The following events were observed when each phone button was used.

Nortel 3905



No EVT_OFFHOOK or EVT_ONHOOK events are observed wih this phone.

1A–1L
EVT_DIGIT_PRESSED
EVT_DIGIT_RELEASED
Subreasons: 0x00000000–0x0000000C

2A EVT_HOLD_BUTTON_PRESSED
EVT_HOLD_BUTTON_RELEASED
2B EVT_RELEASE_BUTTON_PRESSED
EVT_RELEASE_BUTTON_RELEASED

3A–3K EVT_FUNCTION_BUTTON_PRESSED
EVT_FUNCTION_BUTTON_RELEASED
Subreason:
A - 0x0006 B - 0x0005 C - 0x0004
D - 0x0003 E - 0x0002 F - 0x0001
G - 0x0000 H - 0x000B I - 0x000A
J - 0x0009 K - 0x0008

4A-4C EVT_FUNCTION_BUTTON_PRESSED
EVT_FUNCTION_BUTTON_RELEASED
Subreason:
A1 - (Transfer) 0x0011
A2 - (RingAGN) 0x0014
A3 - (PrivRLS) 0x0017
B1 - (Conf) 0x0011
B2 - (Park) 0x0015
B3 - (Charge) 0x0019
C1 - (Forward) 0x0013*
C2 - (Pickup) 0x0016
C3 - (CParty) 0x001A
* The forwarding feature:
Press the Forward button-
EVT_FUNCTION_BUTTON_PRESSED 0x0013
w/ a light flashing event.
Once the extension is entered and accepted, there is a EVT_CFWD.
When the feature is cancelled, there is a EVT_CFWD_CANCELLED

5A-5G
No events have been observed.

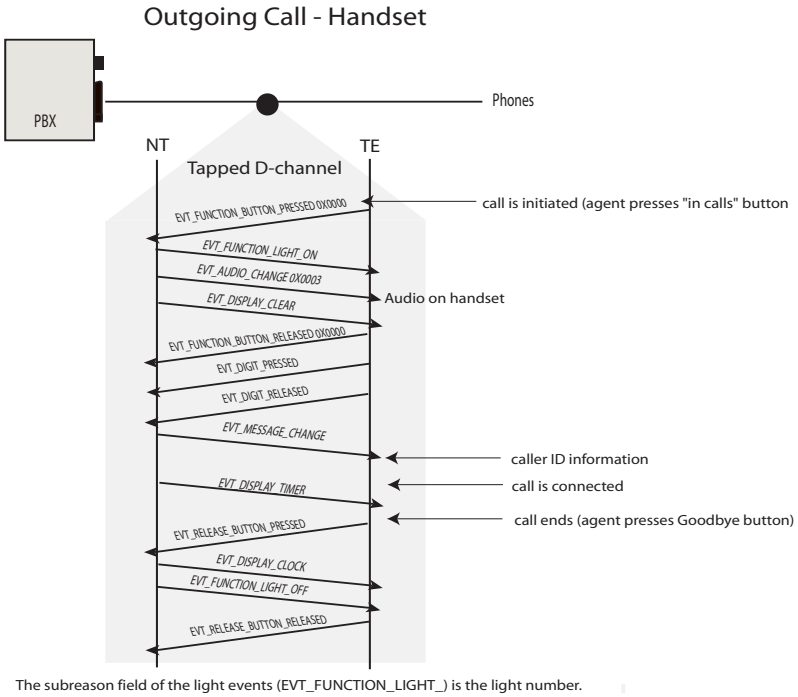
CALL SCENARIOS

The following section shows what events were observed during a typical call scenario. All calls were initiated when the phone was in an idle state:

- When applicable, the handset is ON_HOOK
- All lights are off
- LCD is clear or the clock is displayed

Outgoing Call - Handset

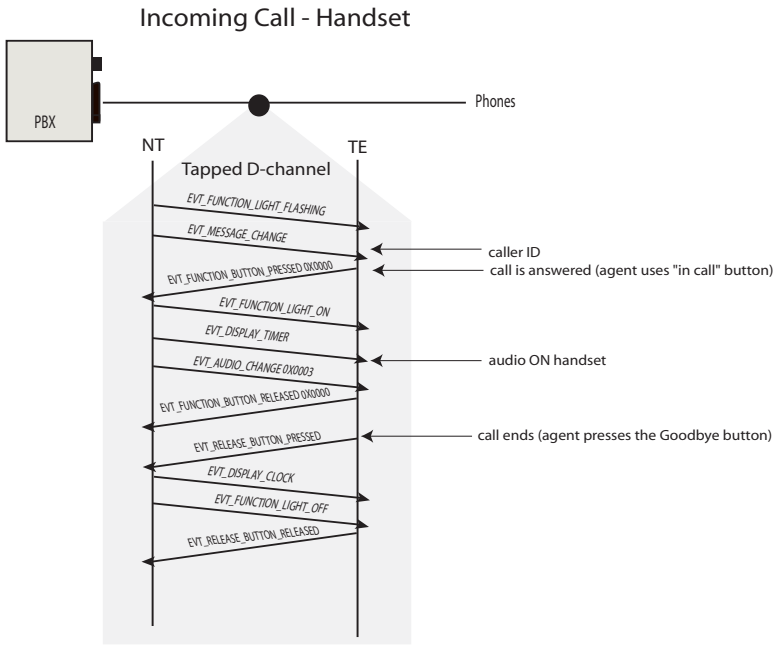
In this scenario the agent captures a line and dials a number. To end the call the releases the call.



The subreason field of the light events (EVT_FUNCTION_LIGHT_) is the light number.

Receiving Call with Handset

An incoming call is received and the agent takes the call. The call is released when the agent replaces the handset.



The subreason field of the light events (EVT_FUNCTION_LIGHT_) is the light number.

Chapter 9

Nortel Norstar & BCM 1

This chapter highlights the use of the SmartWORKS NGX when tapping the Nortel Norstar & BCM.

NOTE: All data in this section was obtained with the Nortel Norstar and Nortel BCM with NT5801FD-93 REL 02B software. If another software version or line card is used, different D-channel patterns may be observed.

Phone Model Support

The following table shows the phone models that have been tested in a tapped environment. This information is updated frequently. For an updated copy refer to the Online Support system's Files and Documents section in the NGX_PBX_MATRIX folder.:

Model	Vox		D-Channel	
7100	✓	T	✓	T
7208	✓	T	✓	T
7316	✓	T	✓	T
7310	✓	T	✓	T
7324	✓	T	✓	T

Status:
T - tested in house
S - supported based on product family (not tested)
R - tested by third party
N - not tested, it may work
W - tested, will not work

Installation and Configuration

Complete installation and wiring diagrams are available in the *SmartWORKS User's Guide*. The following section highlights important installation notes for this particular PBX.

Installation

This PBX is a two wire (2W) model. Follow the standard 2W installation procedures outlined in the *SmartWORKS User's Guide*.

Cable Lengths

AudioCodes USA has extensively tested the cable lengths. Recommendations are available on the Online support system. In the **File and Documents** section look for the **PBX Support Matrix** folder for the *NGX Cable Length Results* document.

CONFIGURATION

After the NGX card has been installed on the digital network, specific configuration steps are required to generate D-channel events.

CLOCK SETTINGS

AudioCodes recommends setting the NGX's clock source to the PBX. This avoids any synchronization problems. Use the ***API MTSetCTMasterClock()*** to set the clock source parameter to NET1. The default setting is to local - the board's clock (OSC).

PBX FIRMWARE

Using the Control Panel, select the **Board** tab. The *PBX Type* field must be set. Use the drop-down menu to select the name of the tapped PBX - Nortel Norstar. Click APPLY (this is required!). The firmware used for this PBX will be automatically installed the next time the board is restarted. Repeat this step for each NGX board and daughter card.

TDM ENCODING

Using the Control Panel, select the **Board** tab. The *TDM Encoding* field must be set. Generally speaking, µ-Law is used in North America and Japan, A-Law is used in Europe. This setting is also PBX dependant. It is highly recommended that you check PBX documentation to verify the setting required on your local network.

By default, the Nortel Norstar & BCM is set to µ-Law. This is a programmable value and AudioCodes recommends checking this setting prior to configuring the NGX.

If this setting is not correct, the recording may sound "metallic".

D-CHANNEL EVENTS

The NGX board does not generate D-channel events by default. This must be enabled. Using the Control Panel select the **Board** tab. The *D-channel* option must be enabled.

EVENT UPDATES

Many PBXs send duplicate commands or phone status reports over the line to the phones. As a result, the NGX decodes this information and generates extraneous events. This can be controlled by disabling the *Event Updates* option in the Control Panel (under Board tab). Once this is disabled, all duplicate events are filtered.

The Nortel Norstar & BCM sends duplicate events and phone status information. If the Event Updates option is disabled in the Control Panel, these events are ignored.

D-Channel Events

The following is a list of all D-channel events reported when tapping Nortel Norstar & BCM PBXs. All events have been grouped by event type.

Results vary depending on the configuration of the PBX in the field, along with the phone model used at the customer site. AudioCodes does not guarantee that all events are reported at each PBX site.

PBX Command Events

The following events are reported from commands passing from the PBX to the phones.

Signalling Events

EVT_RING_OFF
EVT_RING_ON

Audio Events

EVT_AUDIO_CHANGE

Call State Events (All of these events are Meridian specific)

No call state events are reported.

LED (Light) Events

EVT_FUNCTION_LIGHT_FASTFLASHING
EVT_FUNCTION_LIGHT_FLASHING
EVT_FUNCTION_LIGHT_OFF
EVT_FUNCTION_LIGHT_ON

Display (LCD) Events

EVT_DISPLAY_CLEAR
EVT_MESSAGE_CHANGE

Phone (Action) Commands

The following events are reported from data generated by the phone and passed to the PBX.

Hook State Events

EVT_OFFHOOK
EVT_ONHOOK

Button Depression Events

EVT_DIGIT_PRESSED
EVT_FUNCTION_BUTTON_PRESSED
EVT_FEATURE_BUTTON_PRESSED
EVT_HOLD_BUTTON_PRESSED
EVT_RELEASE_BUTTON_PRESSED
EVT_SHIFT_BUTTON_PRESSED
EVT_SOFT_BUTTON_PRESSED

Nortel Norstar & BCM Behavior

Each PBX exhibits unique behaviors. This section shows how common line conditions are handled by the Nortel Norstar & BCM. This section is not meant to be an exhaustive list, but rather an overview of some of the behavior observed by AudioCodes.

NOTE: The following topics are explained in detail in the PBX Integration chapter of this book.

Dialed Numbers (DTMF) Detection

When tapping the Nortel Norstar & BCM the DTMF is not passed in-band. Digits can be detected when local phone buttons are used to dial numbers. This information is decoded from the D-channel and reported as EVT_DIGIT_PRESSED. The exact digit (in ASCII format) is presented to the user application in the subreason field of the MT_EVENT structure.

CallerID

On proprietary PBX networks, callerID is sometimes displayed on phone LCDs. This varies depending on the configuration of the PBX. As a result, AudioCodes cannot guarantee that CallerID is available when tapping the Nortel Norstar & BCM.

When the phone's LCD is updated by the PBX for any reason, the NGX decodes this information and the event EVT_MESSAGE_CHANGE is reported. The ptrBuffer field of the MT_EVENT structure points to the buffer and the "datalength" field is populated with the size of the data buffer pointed to by ptrBuffer, including a null character terminator at the end. **NOTE**: The callerID information must be parsed from the string.

CRC Error Checking

The Nortel Norstar & BCM does not have a CRC check for validating D-channel packets. As a result, when line conditions are poor invalid data can be captured by the NGX and reported to the user application.

CRC errors usually indicate poor line conditions or improper wiring. Because these PBXs do not support CRC checking, it is important to monitor line conditions by using the error checking APIs: ***MTGetFramerStatus()*** and ***MTGetNGXFramerStatistic()***.

PBX Command Events

The following section highlights the observed variations noted with this particular PBX.

Signalling Events - Dial Tone

The Nortel Norstar & BCM generates the dial tone and passes it in-band over the line to the phone. As a result, the Call Progress Monitoring feature is able to detect the signal and the event EVT_MON_DIAL is reported. The signal profiles configured by default on each channel are based on North American standards. The profile parameters of the dial tone must be modified to match the cadence and frequency values of the signal produced by the Nortel Norstar & BCM. Refer to the application notes: *Understanding Signal Profiles* for more information.

Signalling Events - Ring Tones

Nortel Norstar & BCM, sends one EVT_RING_ON and EVT_RING_OFF each time the phone rings. The user application can count the total number of EVT_RING_ON events to determine total ring count.

Phone State Event - Call Forward

At the time of this writing, the Nortel Meridian is the only PBX that generates an event for Call Forwarding (EVT_CFWD and EVT_CFWD_CANCELED). When the agent presses the call forward button to activate this feature, the EVT_CFWD event is reported. A buffer is provided that contains the phone number that the call is forwarded to. When the feature is deactivated by the call agent the event EVT_CFWD_CANCELED is reported.

AUDIO EVENTS - EVT_AUDIO_CHANGE

The PBX controls the audio on the speaker phone, microphone and handset. Should the user enable/disable any of these devices, the PBX will turn on/off the audio for these devices. When this occurs the EVT_AUDIO_CHANGE event is reported to the user application. The subreason field indicates which audio is under control:

Subreason	Description
0x00	Audio Off
0x03	Handset audio receive and transmit
0x08	Handsfree audio receive
0x0C	Handsfree audio receive and transmit

LCD DISPLAY EVENTS

When decoding Nortel Norstar & BCM D-channel, the following LCD events are reported: EVT_MESSAGE_CHANGE, and EVT_DISPLAY_CLEAR. When EVT_MESSAGE_CHANGE is reported the ptrBuffer field of the MT_EVENT structure points to the buffer and the "datalength" field is populated with the size of the data buffer pointed to by ptrBuffer, including a null character terminator at the end. **NOTE**: The callerID information, when available, must be parsed from the string.

LED LIGHT EVENTS

The following light behavior has been observed when using the Nortel Norstar & BCM

EVT_FUNCTION_LIGHT_FASTFLASHING - This has been observed when a call is on hold

EVT_FUNCTION_LIGHT_FLASHING - This has been observed when the phone is ringing

EVT_FUNCTION_LIGHT_ON - a call is in progress, or when a feature is active

EVT_FUNCTION_LIGHT_OFF - phone is idle

Subreason field:

The light subreason field indicates the light number and color. Represented as a hex value the following holds true 0xRRRRCCNN where R = reserved, C = color, and N = light number. The phones that integrate with the Nortel Norstar & BCM do not change colors. Therefore the color bits are not used. The following illustrations shows the bits values set in the subreason field when decoding the Nortel Norstar & BCM:

RRRR	CC	NN
b31-b16	b15-b8	b7-b0
reserved	reserved	Light Number

D-channel events per Phone Model

A complete list of the D-channel events observed when tapping the Nortel Norstar & BCM is provided at the beginning of this chapter. AudioCodes has observed that the types of D-channel events reported may vary per phone model, installation or software version.

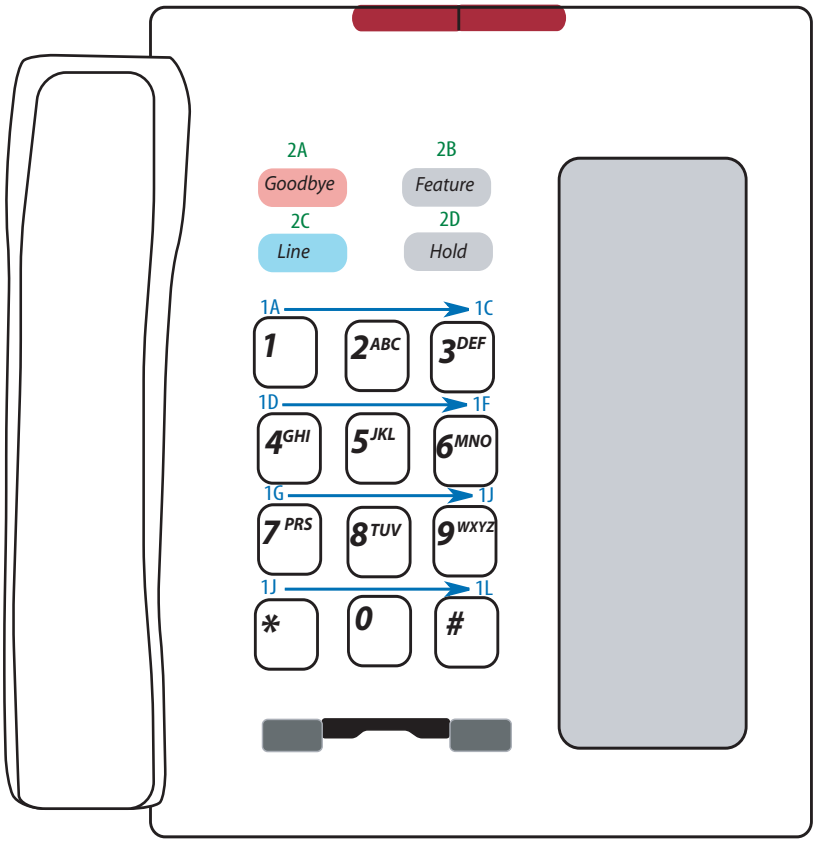
The following section can be used by an application developer to understand variations of D-channel events noted between phone models. This is not meant to be an exhaustive list, but rather an aide to application developers who are getting started.

NOTE: All data in this section was obtained with the Nortel Norstar and Nortel BCM with NT5801FD-93 REL 02B software. If another software version or line card is used, different D-channel patterns may be observed.

PHONE MAP

The following events were observed when each phone button was used.

Meridian 3901



1A–1L
EVT_DIGIT_PRESSED
EVT_DIGIT_RELEASED
Subreasons: 0x00000000–0x0000000C

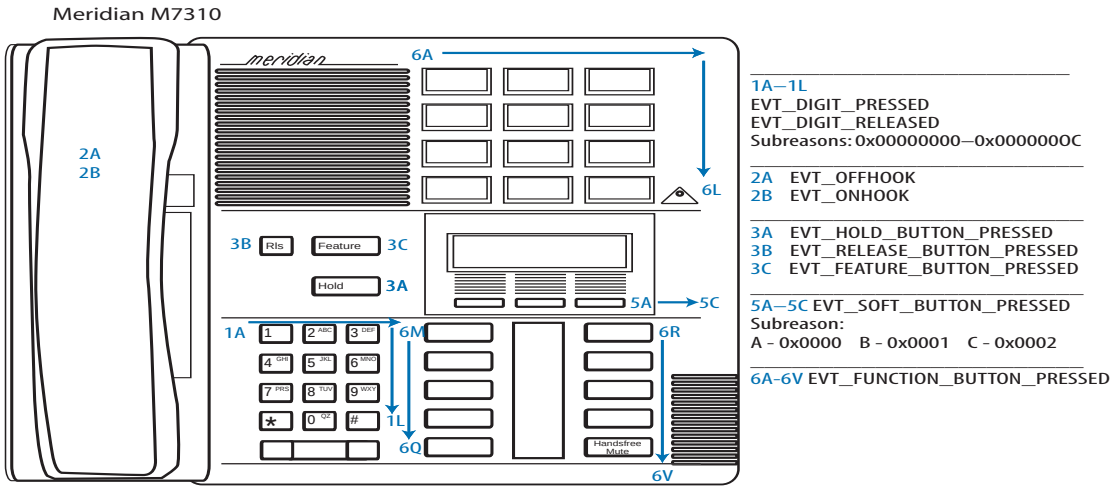
2A EVT_RELEASE_BUTTON_PRESSED
EVT_RELEASE_BUTTON_RELEASED
2B This button acts as a shift key
2C EVT_FUNCTION_BUTTON_PRESSED
EVT_FUNCTION_BUTTON_RELEASED
2D EVT_HOLD_BUTTON_PRESSED
EVT_HOLD_BUTTON_RELEASED

EVT_OFF_HOOK and EVT_ONHOOK events are observed when the handset is used.

M7310

PHONE MAP

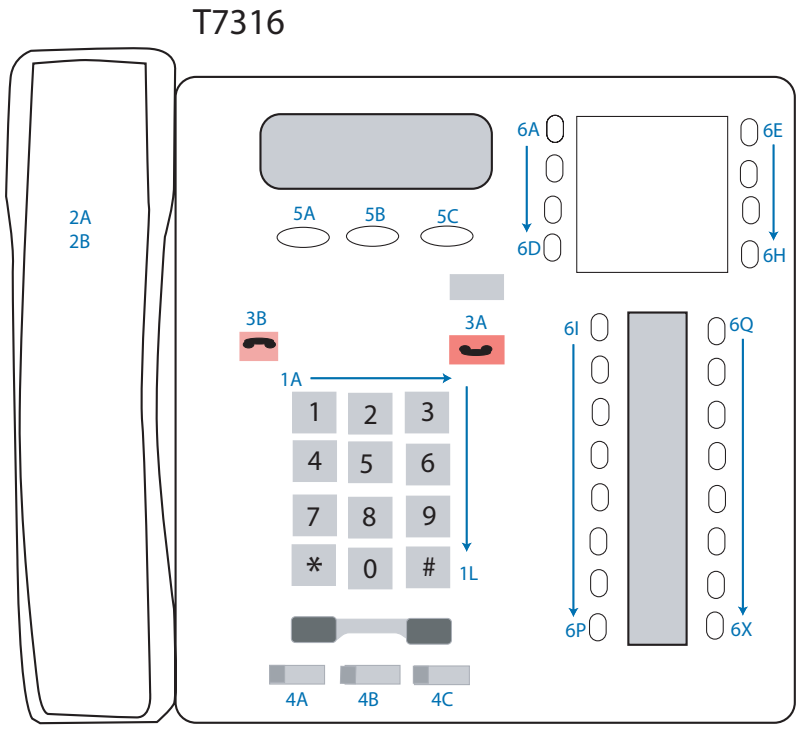
The following events were observed when each phone button was used.



T7316

PHONE MAP

The following events were observed when each phone button was used.



1A–1L
EVT_DIGIT_PRESSED
EVT_DIGIT_RELEASED
Subreasons: 0x00000000–0x0000000C

2A EVT_OFFHOOK
2B EVT_ONHOOK

3A EVT_HOLD_BUTTON_PRESSED
3B EVT_RELEASE_BUTTON_PRESSED

4A–4C EVT_AUDIO_CHANGE
Subreason: (refer to documentation)

5A–5C EVT_SOFT_BUTTON_PRESSED
Subreason:
A - 0x0000 B - 0x0001 C - 0x0002

6A-6X EVT_FUNCTION_BUTTON_PRESSED

M3901

This section provides a detailed look when using this phone by providing a phone map, common call scenarios, and noted behaviors.

CALL SCENARIOS

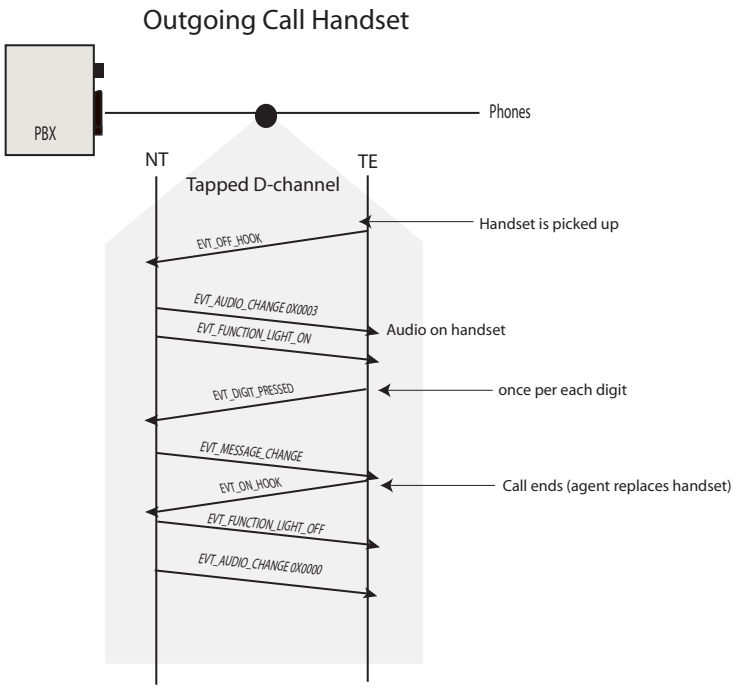
The following section shows what events were observed during a typical call scenario. These call scenarios were not generated by using a specific phone model and should be considered 'generic'. Application developers are encouraged to test with the phone model that will be tapped.

All calls were initiated when the phone is in an idle state:

- When applicable, the handset is ON_HOOK
- All lights are off
- LCD is clear or the clock is displayed.

Outgoing Call - Handset

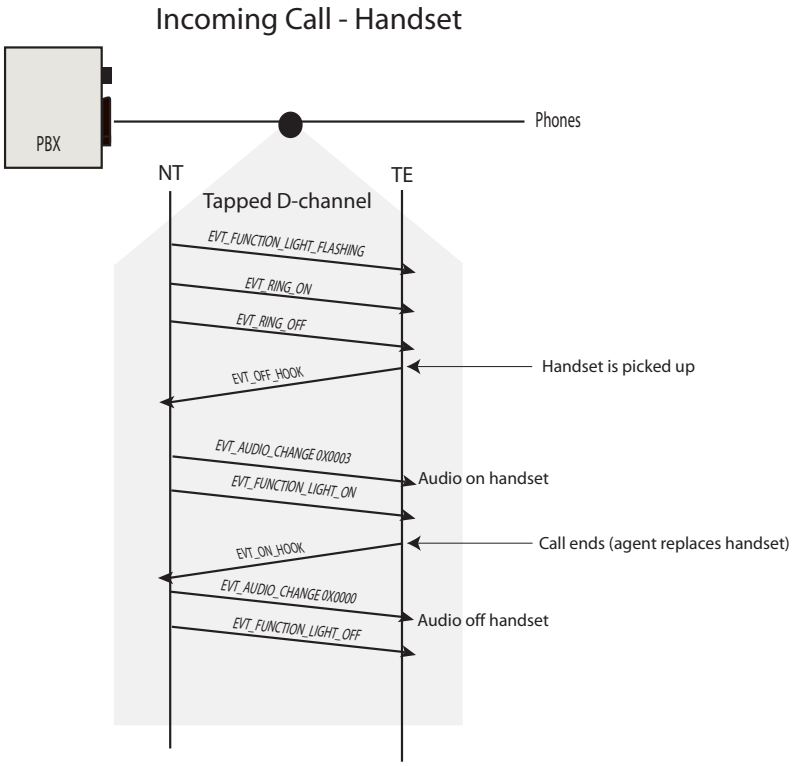
In this scenario the agent picks up the handset and dials a number. To end the call, the agent hangs up the phone by replacing the handset:



The subreason field of the light events (EVT_FUNCTION_LIGHT_) is the light number.

Incoming Call - Handset

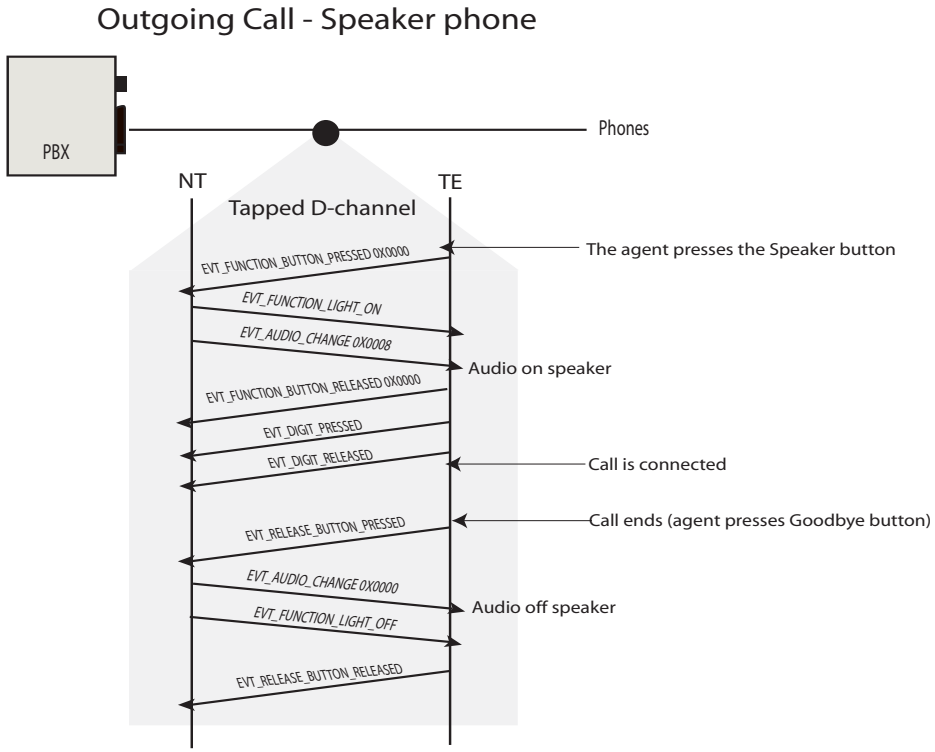
An incoming call is received and the agent takes the call. The call is released when the agent replaces the handset.



The subreason field of the light events (EVT_FUNCTION_LIGHT_) is the light number.

Outgoing Call - Speaker Phone

This call scenario applies specifically to the M7310 phone model. The agent captures a line by pressing the handsfree button and dials the number. The call is terminated when the agent releases the line using the Release button.



The subreason field of the light events (EVT_FUNCTION_LIGHT_) is the light number.

Chapter 10

Panasonic KX

This chapter highlights the use of the SmartWORKS NGX when tapping the Panasonic KX.

Phone Model Support

The following table shows the phone models that have been tested in a tapped environment. This information is updated frequently. For an updated copy refer to the Online Support system's Files and Documents section in the NGX_PBX_MATRIX folder.

Status:
T - tested in house
S - supported based on product family (not tested)
R - tested by third party
N - not tested, it may work
W - tested, will not work:

Model	Vox		D-Channel	
KX-T7431	✓	T	✓	T
KX-T7456	✓	T	✓	T
KX-T7630	✓	T	✓	T
KX-T7636	✓	T	✓	T
KX-T7453	✓	T	✓	T

Installation and Configuration

Complete installation and wiring diagrams are available in the *SmartWORKS User's Guide*. The following section highlights important installation notes for this particular PBX.

INSTALLATION

This PBX can be used as a two wire 2W or 4W scenario, as well as a multi-point environment . Follow the standard 2W and 4W installation procedures outlined in the *SmartWORKS User's Guide*.

When using the T7600 series phones the PBX can be configured for multi-point use (two phones connected per port). The NGX does support this capability with proper installation and application development. Refer to the directions provided in the Multi-Point Installation and Application Development section of this chapter.

Cable Lengths

The following are the maximum cable lengths (in feet) recommended by AudioCodes:

PBX to tap - TBD

tap to NGX - TBD

tap to phones - TBD

CONFIGURATION

After the NGX card has been installed on the digital network, specific configuration steps are required to generate D-channel events.

CLOCK SETTINGS

AudioCodes recommends setting the NGX's clock source to the PBX. This avoids any synchronization problems. Use the ***API MTSetCTMasterClock()*** to set the clock source parameter to NET1. The default setting is to local - the board's clock (OSC).

PBX FIRMWARE

Using the Control Panel, select the **Board** tab. The *PBX Type* field must be set. Use the drop-down menu to select the name of the tapped PBX: *Panasonic KX*. Click APPLY (this is required!). The firmware used for this PBX will be automatically installed the next time the board is restarted. Repeat this step for each NGX board and daughter card. **NOTE:** If tapping a multi-point enviornment, AudioCodes recommends using the Panasonic-S firmware available with release 3.8 or greater.

TDM ENCODING

Using the Control Panel, select the **Board** tab. The *TDM Encoding* field must be set. Generally speaking, µ-Law is used in North America and Japan, A-Law is used in Europe. This setting is also PBX dependant. It is highly recommended that you check PBX documentation to verify the setting required on your local network.

The Panasonic KX can be either A-law or u-law. This is a programmable value and AudioCodes recommends checking this setting prior to configuring the NGX.

If this setting is not correct, the recording may sound "metallic".

D-CHANNEL EVENTS

The NGX board does not generate D-channel events by default. This must be enabled. Using the Control Panel select the **Board** tab. The *D-channel* option must be enabled.

EVENT UPDATES

Many PBXs send duplicate commands or phone status reports over the line to the phones. As a result, the NGX decodes this information and generates extraneous events. This can be controlled by disabling the *Event Updates* option in the Control Panel (under Board tab). Once this is disabling, all duplicate events are filtered.

The Panasonic KX do not send phone status information. However they may send duplicate commands. It is best to disable this field so that all duplicate events can be filtered by the NGX.

Multi-Point Installation and Application Development

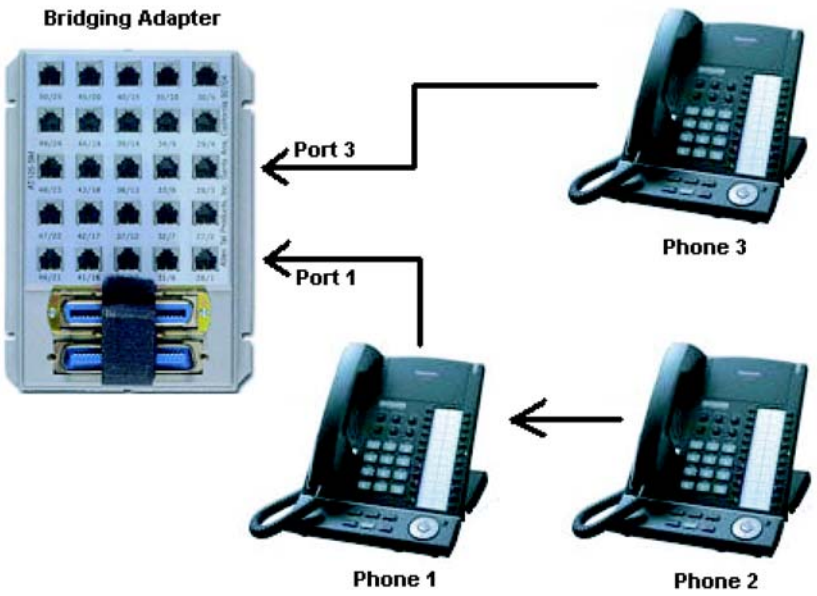
The Panasonic KX can be configured for multi-point (multiple phones connected to a single port) when using the T7600 series of phones. The following installation and application development guidelines explain how to use the NGX to tap a multi-point environment.

INSTALLATION

On any PBX system the D-channel data for all phones is carried along the same physical interface while separate B channels are used to transmit the voice data(B1 & B2). When using a breakout box to tap the network, ports must be skipped when two phones are connected together. For instance, phone 2 will be connected to

phone 1, phone 1 will be connect to port 1. Phone 3 will be connected to port 3. This will eliminate confusion when separating the B-Channels into separate channels on the NGX when recording.

The following diagram illustrates this:



CONFIGURING THE NGX

As of release 3.8, the SmartWORKS SDK offers two Panasonic firmwares: Panasonic and Panasonic - S. Both firmwares support multi-point tapping, however the Panasonic firmware requires the user to modify their tapping appliation. The Panasonic - S firmware does not require application development.

APPLICATION DEVELOPMENT

The Panasonic firmware does not accommodate the multi-point enviroment from the perspective of the user's application. As a result, the user application must be modified to properly handle Dchannel events and voice recording. The Panasonic-S firmware (supported with 3.8 or greater) manages Dchannel event reporting and channel resources so that the user application does not have to be modified. Using the above diagram, lets take a look at how the D-channel and voice data is presented to the user application when using the Panasonic Firmware.

In a typical installation, the B1 voice data is connected to a single channel's DSP resources on the NGX. The B2 voice data is not connected to anything by default. On a multi-point system the B channel for phone 1 is processed by the DSP resource allocated for that channel on the NGX. The B channel for phone 2 must be passed via the TDM bus so that another NGX channel resource can be used for signal processing and recording. The D-channel for both phones is processed by the first channel's resources on the NGX and D-channel events for both phones are reported on the same channel queue. Using the callback function, your application can be designed so that all D-channel events for phone 2 are passed over to the second channel's queue. Refer to the **DChanMultiTerm** sample application available on the online support system if your application is using the Panasonic firmware.

D-Channel Event Reporting

The NGX is decoding all D-channel packets for both phones on the same channel resource and reporting all D-channel events on the same channel queue. Using the event callback function, the user application must first determine if the event is a D-channel event. If yes, then the user application can determine whether the event is for phone 1 or 2. This information is presented in the *XtraInfo* field of the MT_EVENT structure. (phone 1 = 0x0001 and phone 2 = 0x0002). If the event is associated with phone 2, the user application can then invoke the function ***MTPutChannelEvent()*** to put this event into channel 2's event queue. From here, all D-channel events for phone 2 can be processed by your application and any start/stop recording triggers will be honored.

Managing B - Channel Data

The voice data for both phones are transmitted on separate B-channels (B1 and B2 respectively). Looking at the above illustration, the voice data for phone 1 is recorded using the channel resource associated with channel 1 on the NGX. To access the B2 voice data the user's application must rely on the ***MTSetFramerOutput()*** API to pull the B2 voice data to the TDM bus. Then the application must open another channel (channel 2) on the NGX where it's inputs are connected to the TDM bus (***MTSetInputs()***). **NOTE:** The voice data put onto the TDM bypasses DSP resources on the first channel, therefore all gain and detection settings must be managed on the second channel.

D-Channel Events

The following is a list of all D-channel events reported when tapping Panasonic KX PBXs. All events have been grouped by event type.

Results vary depending on the configuration of the PBX in the field, along with the phone model used at the customer site. AudioCodes does not guarantee that all events are reported at each PBX site.

NOTE: The following topics are explained in detail in the PBX Integration chapter of this book.

PBX Command Events

The following events are reported from commands passing from the PBX to the phones.

Signalling Events

EVT_RING_OFF
EVT_RING_ON

Audio Events

EVT_AUDIO_CHANGE

LED (Light) Events

EVT_CONF_LIGHT_FASTFLASHING
EVT_CONF_LIGHT_FLASHING
EVT_CONF_LIGHT_ON
EVT_CONF_LIGHT_OFF
EVT_CONF_LIGHT_QUICKFLASH
EVT_DND_LIGHT_FASTFLASHING
EVT_DND_LIGHT_FLASHING
EVT_DND_LIGHT_ON
EVT_DND_LIGHT_OFF
EVT_DND_LIGHT_QUICKFLASH
EVT_HANDSFREE_LIGHT_FASTFLASHING
EVT_HANDSFREE_LIGHT_FLASHING
EVT_HANDSFREE_LIGHT_ON
EVT_HANDSFREE_LIGHT_OFF
EVT_HANDSFREE_LIGHT_QUICKFLASH
EVT_INTERCOM_LIGHT_FASTFLASHING
EVT_INTERCOM_LIGHT_FLASHING
EVT_INTERCOM_LIGHT_ON
EVT_INTERCOM_LIGHT_OFF
EVT_INTERCOM_LIGHT_QUICKFLASH
EVT_LINE_LIGHT_FASTFLASHING
EVT_LINE_LIGHT_FLASHING
EVT_LINE_LIGHT_ON
EVT_LINE_LIGHT_OFF
EVT_LINE_LIGHT_QUICKFLASH
EVT_MESSAGE_LIGHT_FASTFLASHING
EVT_MESSAGE_LIGHT_FLASHING
EVT_MESSAGE_LIGHT_ON
EVT_MESSAGE_LIGHT_OFF
EVT_MESSAGE_LIGHT_QUICKFLASH
EVT_MUTE_LIGHT_FASTFLASHING
EVT_MUTE_LIGHT_FLASHING
EVT_MUTE_LIGHT_ON
EVT_MUTE_LIGHT_OFF
EVT_MUTE_LIGHT_QUICKFLASH
EVT_RING_LIGHT_FASTFLASHING
EVT_RING_LIGHT_FLASHING
EVT_RING_LIGHT_ON
EVT_RING_LIGHT_OFF
EVT_RING_LIGHT_QUICKFLASH
EVT_SPEEDIAL_LIGHT_FASTFLASHING
EVT_SPEEDIAL_LIGHT_FLASHING

EVT_SPEEDIAL_LIGHT_ON
EVT_SPEEDIAL_LIGHT_OFF
EVT_SPEEDIAL_LIGHT_QUICKFLASH

Display (LCD) Events

EVT_DISPLAY_CLEAR
EVT_MESSAGE_CHANGE

Call State Events

No call state events are reported for this PBX.

Phone (Action) Commands

The following events are reported from data generated by the phone and passed to the PBX.

Hook State Events

EVT_OFFHOOK
EVT_ONHOOK

Button Depression Events

EVT_DIGIT_PRESSED
EVT_FUNCTION_BUTTON_PRESSED
EVT_CANCEL_BUTTON_PRESSED
EVT_CONF_BUTTON_PRESSED
EVT_DND_BUTTON_PRESSED
EVT_ENTER_BUTTON_PRESSED
EVT_FLASH_BUTTON_PRESSED
EVT_FLASH_BUTTON_RELEASED
EVT_HOLD_BUTTON_PRESSED
EVT_INTERCOM_BUTTON_PRESSED
EVT_LINE_BUTTON_PRESSED
EVT_MESSAGE_BUTTON_PRESSED
EVT_MODE_BUTTON_PRESSED
EVT_MUTE_BUTTON_PRESSED
EVT_PAUSE_BUTTON_PRESSED
EVT_REDIAL_BUTTON_PRESSED
EVT_SELECT_BUTTON_PRESSED
EVT_SHIFT_BUTTON_PRESSED
EVT_SOFT_BUTTON_PRESSED
EVT_SPEAKER_BUTTON_PRESSED
EVT_SPEEDIAL_BUTTON_PRESSED
EVT_TRANSFER_BUTTON_PRESSED

Panasonic KX Behavior

Each PBX exhibits unique behaviors. This section shows how common line conditions are handled by the Panasonic KX. This section is not meant to be an exhaustive list of PBX behavior, but rather an aid to developers who are getting started.

NOTE: The following topics are explained in detail in the PBX Integration chapter of this book.

Dialed Numbers (DTMF) Detection

When tapping a Panasonic KX, DTMF tones are transmitted as in-band signals and messages are passed in the D-channel. When a digit is pressed by the call agent, this is communicated to the PBX via a D-channel message and is reported as EVT_DIGIT_PRESSED. The PBX does reply with in-band DTMF signals, however these are not detected by the NGX.

CallerID

On proprietary PBX networks, CallerID is sometimes displayed on phone LCDs. This varies depending on the configuration of the PBX. As a result, AudioCodes cannot guarantee that CallerID is available when tapping the Panasonic KX.

When the phones LCD is updated by the PBX for any reason, the NGX decodes this information and the event EVT_MESSAGE_CHANGE is reported. The ptrBuffer field of the MT_EVENT structure points to the buffer and the "datalength" field is populated with the size of the data buffer pointed to by ptrBuffer, including a null character terminator at the end. **NOTE**: The CallerID information must be parsed from this string.

CRC Error Checking

The Panasonic KXrelies on a 16-bit CRC check. Invalid messages are not decoded, and error message are logged in the system event viewer.

PBX Command Events

The following section highlights the observed variations noted with this particular PBX.

Signalling Events - Dial Tone

The Panasonic KX produces the dial tone and it is passed in-band to the telephone. This is detected by the Call Progress Monitoring (CPM) capabilities of the NGX and reported as EVT_MON_DIAL. Refer to Call Progress Monitoring application notes for more information about CPM.

Signalling Events - Ring Tones

Panasonic KX, sends one command to alert the phone of an incoming call. The phone generates the ring tone. This is reported as EVT_RING_ON. When the call is answered by the agent, the PBX sends the command to stop ringing. This is reported as EVT_RING_OFF. The user application must rely on the timestamp between the two events to determine how long the phone has been ringing.

Ring Cadence

The Panasonic KX produces a dual ring tone which is programmable. When the EVT_RING_ON/ EVT_RING_OFF event is reported, the cadence value of the tone is reported to the user application in the subreason field (0xC2C1) where 0xC1 is cadence of tone 1, and 0xC2 is cadence of tone 2. Thirty different dual ring tone values have been seen. When a single ring tone is used the value of 0xC1 is the ring type and the sound is similar to the 74XX phones. The 0xC2 is the pitch of the tone.

LCD Display Events

EVT_DISPLAY_CLEAR, and EVT_MESSAGE_CHANGE are reported when tapping the Panasonic KX PBXs.

EVT_AUDIO_CHANGE

The EVT_AUDIO_CHANGE indicates the state of the audio on the phone. The subreason field of this event is used to identify the state of the device (on or off) as well as the device that is under control (speaker, handset or microphone). Refer to the chapter the explains PBX Integration for details on the subreason field.

Because the PBX does not have control of the handset audio path completely, the NGX is unable to report the status of the handset. No audio change events are reported relative to the audio path of the handset. The phone's speaker is controlled by the PBX completely and it's state is reported by the NGX.

LED or Light events

Subreason field:

The light subreason field indicates the light number and color. Represented as a hex value the following holds true 0xRRRRCCNN where R = reserved, C = color, and N = light number. The lights on the Panasonic KX can blink red or green, therefore the light color is specified with bits 8-15. Light number 1, green would be represented as: 0x0101. Light number 10, red would be represented as 0x020A.

The following illustration represents the bits of the subreason field used when decoding the Panasonic KX:

RRRR	CC			NN
b31-b16	b15-b10	b9	b8	b7-b0
reserved	reserved	Red	Green	Light Number

Mute/Auto Answer LED

This LED is dual-purpose. When a line is active on the phone, this LED acts as a mute light. When all lines are idle this acts as an auto-answer LED. In all cases the NGX reports this LED as a EVT_MUTE_LIGHT_.....

When LED is acting as a mute light, then it is possible to combine these events with hook events (EVT_ONHOOK and EVT_OFFHOOK) to determine the handset audio transmission path.

For example: When the OFF_HOOK event is reported and the mute LED is ON, then the user application can interpret this the same as EVT_AUDIO_CHANGE with handset received active and handset transmit inactive.

Message LED

At times the EVT_MESSAGE_LIGHT_FLASHING is reported, though the actual LED on the phone is on solid.

Intercom LED

The intercom LED has only been seen as green when testing in the AudioCodes lab.

D-channel events per Phone Model

A complete list of the D-channel events observed when tapping the Panasonic KX is provided at the beginning of this chapter. AudioCodes has observed that the types of D-channel events reported may vary per phone model, installation or software version.

The following section can be used by an application developer to understand the variations noted between phone models. This is not meant to be an exhaustive list, but rather an aide to application developers who are getting started.

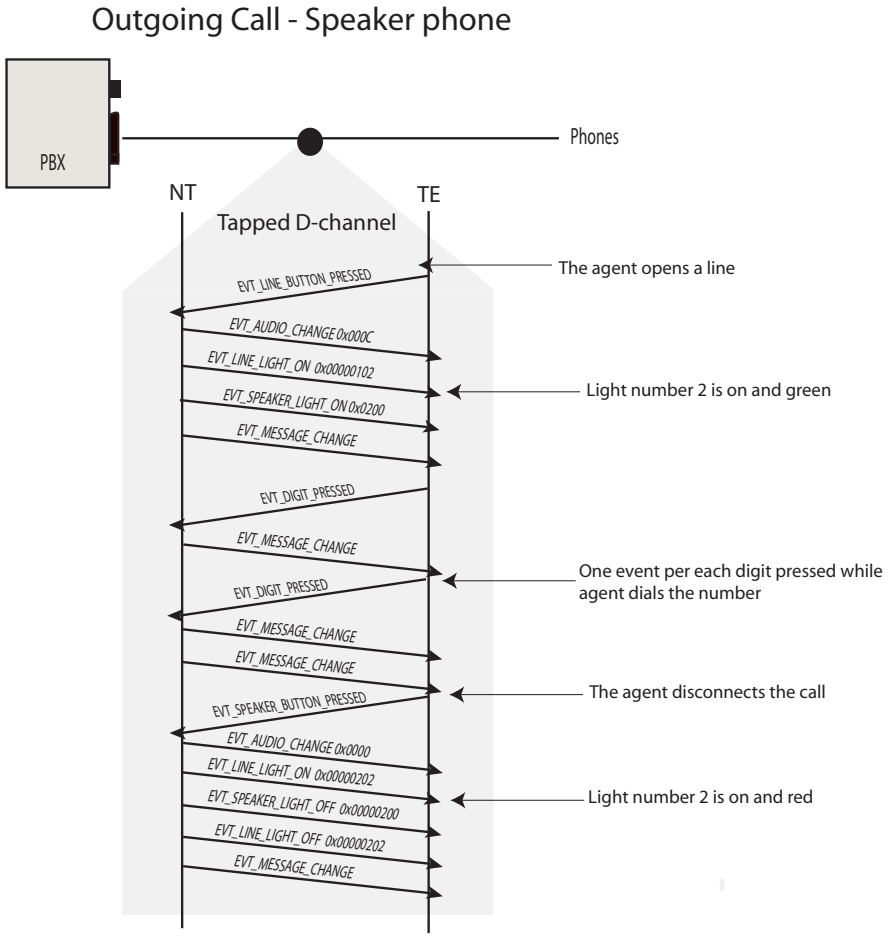
KX-T7431

CALL SCENARIOS

The following section shows what events were observed during a typical call scenario:

Outgoing Call - Speaker Phone

The agent captures a line and dials the number. The call is terminated by the call agent.

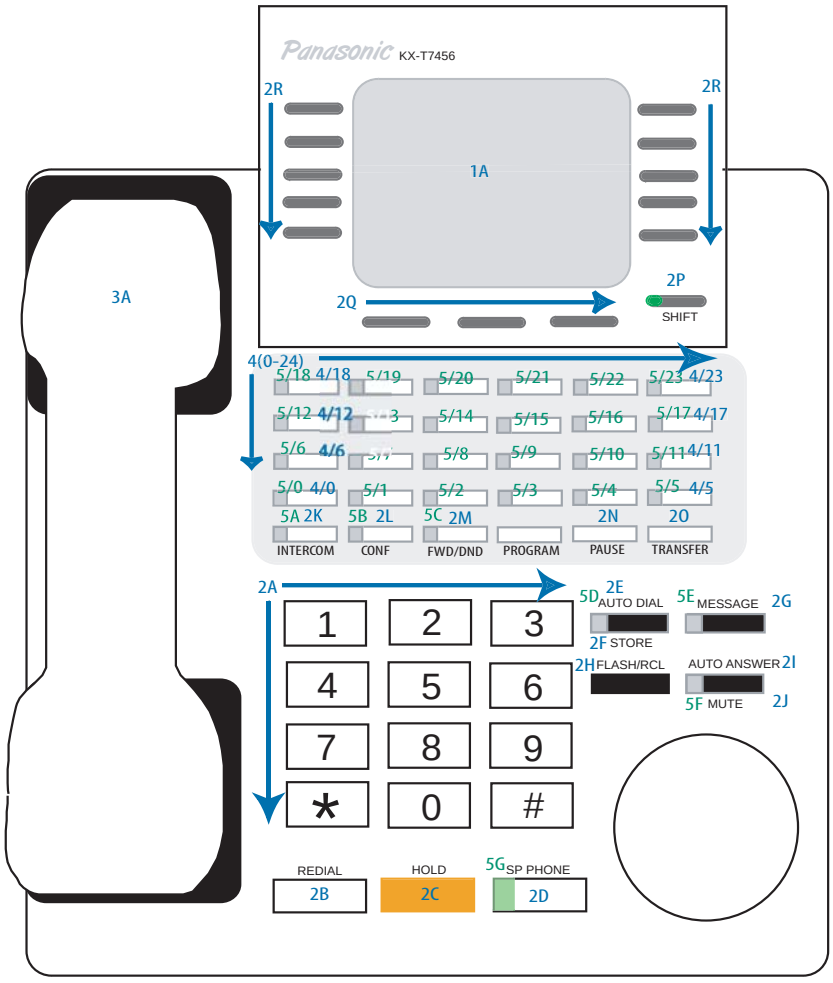


KX-T7456

PHONE MAP

The following events were observed when each phone button was used.

Panasonic KX-T7456



1A
EVT_MESSAGE_CJHANGE
EVT_DISPLAY_CLEAR

2A EVT_DIGIT_BUTTON_PRESSED
2B EVT_REDIAL_BUTTON_PRESSED
2C EVT_HOLD_BUTTON_PRESSED
2D EVT_SPEAKER_BUTTON_PRESSED
2E/2F EVT_SPEEDDIAL_BUTTON_PRESSED
2G EVT_MESSAGE_BUTTON_PRESSED
2H EVT_FLASH_BUTTON_PRESSED
EVT_FLASH_BUTTON_RELEASED
2I/2J EVT_MUTE_BUTTON_PRESSED
2K EVT_INTERCOM_BUTTON_PRESSED
2L EVT_CONF_BUTTON_PRESSED
2M EVT_DND_BUTTON_PRESSED
(when call forwarding is enabled, DND is reported)
2N EVT_PAUSE_BUTTON_PRESSED
2O EVT_TRANSFER_BUTTON_PRESSED
2P EVT_SHIFT_BUTTON_PRESSED
2Q EVT_SOFT_BUTTON_PRESSED
2R EVT_FUNCTION_BUTTON_PRESSED

3A EVT_OFFHOOK
EVT_ONHOOK

4(0-23) EVT_LINE_BUTTON_PRESSED

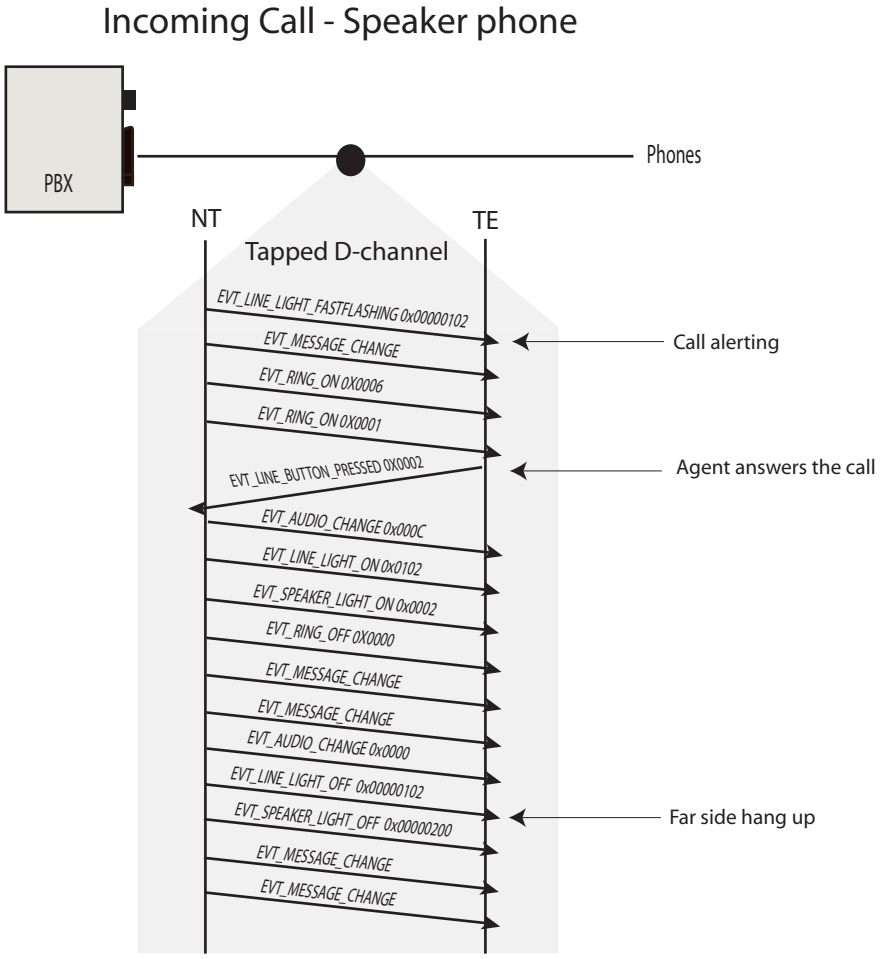
LED Events (ON/OFF/FASTFLASHING/
FLASHING/QUICKFLASH)
5(0-23) EVT_LINE_LIGHT_
(Subreason field indicates light number and color)
5A EVT_INTERCOM_LIGHT_
5B EVT_CONF_LIGHT_
5C EVT_DND_LIGHT_
5D EVT_SPEEDDIAL_LIGHT_
5E EVT_HANDSFREE_LIGHT_

CALL SCENARIOS

The following section shows what events were observed during a typical call scenario:

Incoming Call - Speaker Phone

The agent phone is alerted of an incoming call and is answered using the speaker. The far side disconnects the call:

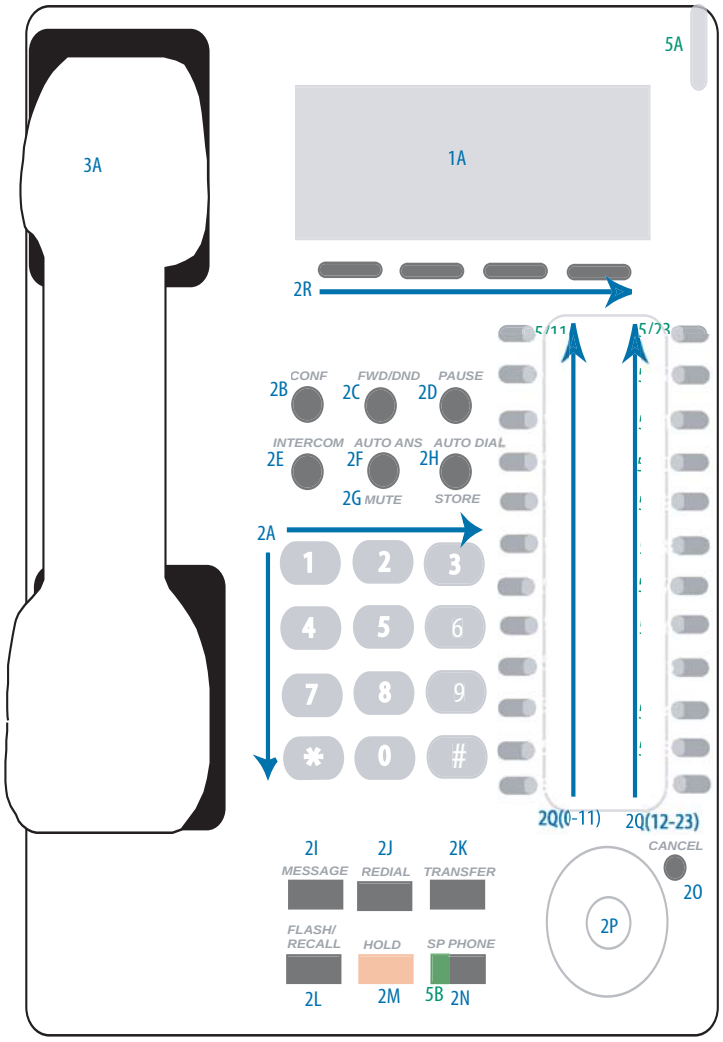


KX-T7636

Phone Map

The following events were observed when each phone button was used.

Panasonic KX-T7636



1A
EVT_MESSAGE_CJHANGE
EVT_DISPLAY_CLEAR

2A EVT_DIGIT_BUTTON_PRESSED
2B EVT_CONF_BUTTON_PRESSED
2C EVT_DND_BUTTON_PRESSED
(When call forwarding is enabled, DND event is reported)
2D EVT_PAUSE_BUTTON_PRESSED
2E EVT_INTERCOM_BUTTON_PRESSED
2F/2G EVT_MUTE_BUTTON_PRESSED
2H EVT_SPEEDDIAL_BUTTON_PRESSED
2I EVT_MESSAGE_BUTTON_PRESSED
2J EVT_REDIAL_BUTTON_PRESSED
2K EVT_TRANSFER_BUTTON_PRESSED
2L EVT_FLASH_BUTTON_PRESSED
EVT_FLASH_BUTTON_RELEASED
2M EVT_HOLD_BUTTON_PRESSED
2N EVT_SPEAKER_BUTTON_PRESSED
2O EVT_CANCEL_BUTTON_PRESSED
2P EVT_ENTER_BUTTON_PRESSED
2Q EVT_LINE_BUTTON_PRESSED
(The subreason field indicates line number)
2R EVT_SOFT_BUTTON_PRESSED

3A EVT_OFFHOOK
EVT_ONHOOK

4(0-24) EVT_LINE_BUTTON_PRESSED

LED Events (ON/OFF/FASTFLASHING/
FLASHING/QUICKFLASH)
5(0-23) EVT_LINE_LIGHT_
(Subreason field indicates light number and color)
5A EVT_RING_LIGHT_
5B EVT_HANDSFREE_LIGHT_
Subreason: 0x00000000
Bits 0-7 = Light Number
Bits 8 = Green
Bit 9 = Red

Chapter 11

Siemens HiPath & HiCom

This chapter highlights the use of the SmartWORKS NGX when tapping the Siemens HiPath & HiCom.

NOTE: All data in this section was obtained with the following PBX software:

HiCom 100 E & 150E - not documented

HiCom 300 E - 9006.6 SMR1 SMPE software and Q2473-X SLMO and Q2158-X SLMO line cards

HiPath 3000 & 3750 - HE580S.04.517 Version 4 and Q2901-X SLMO line card

HiPath 4000 - SMR10 SMP4 Version 1.0 and Q2168-X SLMO line card

Phone Model Support

The following table shows the phone models that have been tested in a tapped environment. This information is updated frequently. For an updated copy refer to the Online Support system's Files and Documents section in the NGX_PBX_MATRIX folder.

Status:
T - tested in house
S - supported based on product family (not tested)
R - tested by third party
N - not tested, it may work
W - tested, will not work

HiCom 100E and 150E:

Model	Vox		D-Channel	
Optiset Basic	✓	T	✓	T
Optiset Standard	✓	T	✓	T
Optiset Advanced	✓	T	✓	T
Optipoint 500 entry	✓	S	✓	S
Optipoint 500 economy	✓	S	✓	S
Optipoint 500 basic	✓	S	✓	S
Optipoint 500 standard	✓	T	✓	T
Optipoint 500 advanced	✓	S	✓	S

HiCom 300E:

Model	Vox		D-Channel	
Optiset Basic	✓	T	✓	T
Optiset Standard	✓	T	✓	T
Optiset Advanced	✓	T	✓	T
Optipoint 500 entry	✓	T	✓	T
Optipoint 500 economy	✓	T	✓	T
Optipoint 500 basic	✓	T	✓	T
Optipoint 500 standard	✓	T	✓	T
Optipoint 500 advanced	✓	T	✓	T

HiCom 3000 and 3750:

Model	Vox		D-Channel	
Optiset Basic	✓	T	✓	T
Optiset Standard	✓	T	✓	T
Optiset Advanced	✓	T	✓	T
Optipoint 500 entry	✓	T	✓	T
Optipoint 500 economy	✓	T	✓	T
Optipoint 500 basic	✓	T	✓	T
Optipoint 500 standard	✓	T	✓	T
Optipoint 500 advanced	✓	T	✓	T

HiCom 4000:

Model	Vox		D-Channel	
Optiset Basic	✓	T	✓	T
Optiset Standard	✓	T	✓	T
Optiset Advanced	✓	T	✓	T
Optipoint 500 entry	✓	T	✓	T
Optipoint 500 economy	✓	T	✓	T
Optipoint 500 basic	✓	T	✓	T
Optipoint 500 standard	✓	T	✓	T
Optipoint 500 advanced	✓	T	✓	T
Advanced Conference DE	✓	T	✓	T

Installation and Configuration

Complete installation and wiring diagrams are available in the *SmartWORKS User's Guide*. The following section highlights important installation notes for this particular PBX.

Installation

This PBX is a two wire (2W) model. Follow the standard 2W installation procedures outlined in the *SmartWORKS User's Guide*.

Cable Lengths

The following are the maximum cable lengths (in feet) recommended by AudioCodes:

HiCom - 100E, 150E, and 300E

PBX to tap - 75'

tap to NGX - 300'

tap to phones - 1545'

HiPath 3000, 3750, and 4000

PBX to tap - 75'

tap to NGX - 300'

tap to phones - 1545'

CONFIGURATION

After the NGX card has been installed on the digital network, specific configuration steps are required to generate D-channel events.

CLOCK SETTINGS

AudioCodes recommends setting the NGX's clock source to the PBX. This avoids any synchronization problems. Use the ***API MTSetCTMasterClock()*** to set the clock source parameter to NET1. The default setting is to local - the board's clock (OSC).

PBX FIRMWARE

Using the Control Panel, select the **Board** tab. The *PBX Type* field must be set. Use the drop-down menu to select the name of the tapped PBX: *Siemens HiCom/HiPath*. Click APPLY (this is required!). The firmware used for this PBX will be automatically installed the next time the board is restarted. Repeat this step for each NGX board and daughter card.

TDM ENCODING

Using the Control Panel, select the **Board** tab. The *TDM Encoding* field must be set. Generally speaking, µ-Law is used in North America and Japan, A-Law is used in Europe. This setting is also PBX dependant. It is highly recommended that you check PBX documentation to verify the setting required on your local network.

By default, the Siemens HiPath is set to A-Law and the Siemens HiCom is set to µ-Law. This is a programmable value and AudioCodes recommends checking this setting prior to configuring the NGX.

If this setting is not correct, the recording may sound "metallic".

D-CHANNEL EVENTS

The NGX board does not generate D-channel events by default. This must be enabled. Using the Control Panel select the **Board** tab. The *D-channel* option must be enabled.

EVENT UPDATES

Many PBXs send duplicate commands or phone status reports over the line to the phones. As a result, the NGX decodes this information and generates extraneous events. This can be controlled by disabling the *Event Updates* option in the Control Panel (under Board tab). Once this is disabling, all duplicate events are filtered.

The Siemens HiPath & HiCom do not send phone status information. However they may send duplicate commands. It is best to disable this field so that all duplicate events can be filtered by the NGX.

D-Channel Events

The following is a list of all D-channel events reported when tapping Siemens HiPath & HiCom PBXs. All events have been grouped by event type.

Results vary depending on the configuration of the PBX in the field, along with the phone model used at the customer site. AudioCodes does not guarantee that all events are reported at each PBX site.

NOTE: The following topics are explained in detail in the PBX Integration chapter of this book.

PBX Command Events

The following events are reported from commands passing from the PBX to the phones.

Signalling Events

EVT_RING_OFF
EVT_RING_ON

Audio Events

EVT_AUDIO_CHANGE

LED (Light) Events

EVT_FUNCTION_LIGHT_FASTFLASHING
EVT_FUNCTION_LIGHT_FLASHING
EVT_FUNCTION_LIGHT_ON
EVT_FUNCTION_LIGHT_OFF
EVT_FUNCTION_LIGHT_QUICKFLASH
EVT_FUCNTION_LIGHT_VERY_FASTFLASHING

Display (LCD) Events

EVT_DISPLAY_CLEAR
EVT_DISPLAY_CLOCK
EVT_MESSAGE_CHANGE

Call State Events

No call state events are reported for this PBX.

Phone (Action) Commands

The following events are reported from data generated by the phone and passed to the PBX.

Hook State Events

EVT_OFFHOOK
EVT_ONHOOK

Button Depression Events

EVT_DIGIT_PRESSED
EVT_FUNCTION_BUTTON_PRESSED
EVT_FUNCTION_BUTTON_RELEASED
EVT_SOFT_BUTTON_PRESSED

Siemens HiPath & HiCom Behavior

Each PBX exhibits unique behaviors. This section shows how common line conditions are handled by the Siemens HiPath & HiCom. This section is not meant to be an exhaustive list of PBX behavior, but rather an aid to developers who are getting started.

NOTE: The following topics are explained in detail in the PBX Integration chapter of this book.

Dialed Numbers (DTMF) Detection

When tapping a Siemens HiPath & HiCom, DTMF is detected both ways. In-band DTMF signals are detected by the NGXs DTMF detectors and reported as EVT_DIGIT. When the agent dials a number, this data is passed to the PBX on the D-channel. This is reported as EVT_DIGIT_PRESSED. In both cases, the actual digit (in ASCII format) is passed to the user application in the sub reason field of the MT_EVENT structure.

CallerID

On proprietary PBX networks, callerID is sometimes displayed on phone LCDs. This varies depending on the configuration of the PBX. As a result, AudioCodes cannot guarantee that CallerID is available when tapping the Siemens HiPath & HiCom.

When the phones LCD is updated by the PBX for any reason, the NGX decodes this information and the event EVT_MESSAGE_CHANGE is reported. The ptrBuffer field of the MT_EVENT structure points to the buffer and the "datalength" field is populated with the size of the data buffer pointed to by ptrBuffer, including a null character terminator at the end. **NOTE**: The callerID information must be parsed from the string.

CRC Error Checking

The Siemens HiPath & HiComrelies on a 16-bit CRC check. Invalid messages are not decoded, and error message are logged in the system event viewer.

PBX Command Events

The following section highlights the observed variations noted with this particular PBX.

Signalling Events - Dial Tone

The Siemens HiPath & HiCom passes a command to the phone when a dial tone must be present. This command is not decoded by the NGX. No events are reported.

Signalling Events - Ring Tones

Siemens HiPath & HiCom, sends one command to alert the phone of an incoming call. The phone generates the ring tone. This is reported as EVT_RING_ON. When the call is answered by the agent, the PBX sends the command to stop ringing. This is reported as EVT_RING_OFF. The user application must rely on the timestamp between the two events to determine how long the phone has been ringing.

LCD Display Events

EVT_DISPLAY_CLEAR, EVT_DISPLAY_CLOCK, and EVT_MESSAGE_CHANGE are reported when tapping the Siemens HiPath & HiCom PBXs.

EVT_AUDIO_CHANGE

The EVT_AUDIO_CHANGE indicates the state of the audio on the phone. The subreason field of this event is used to identify the state of the device (on or off) as well as the device that is under control (speaker, handset or microphone). Refer to the chapter the explains PBX Integration for details on the subreason field.

LED OR LIGHT EVENTS

EVT_FUNCTION_LIGHT_OFF - line is idle

EVT_FUNCTION_LIGHT_ON - active call

EVT_FUNCTION_LIGHT_VERY_FASTFLASHING - line is mute

EVT_FUNCTION_LIGHT_FAST_FLASHING - line is ringing

EVT_FUNCTION_LIGHT_FLASHING - line is on hold

Subreason field:

The light subreason field indicates the light number and color. Represented as a hex value the following holds true 0xRRRRCCNN where R = reserved, C = color, and N = light number. The lights on both the Siemens HiPath & HiCom do not change color, therefore the color bits are not used. The following illustration represents the bits of the subreason field used when decoding the Siemens HiPath & HiCom:

RRRR	CC	NN
b31-b16	b15-b8	b7-b0
reserved	reserved	Light Number

D-channel events per Phone Model

A complete list of the D-channel events observed when tapping the Siemens HiPath & HiCom is provided at the beginning of this chapter. AudioCodes has observed that the types of D-channel events reported may vary per phone model, installation or software version.

The following section can be used by an application developer to understand the variations noted between phone models. This is not meant to be an exhaustive list, but rather an aide to application developers who are getting started.

NOTE: All data in this section was obtained with the following PBX software:

HiCom 100 E & 150E - not documented

HiCom 300 E - 9006.6 SMR1 SMPE software and Q2473-X SLMO and Q2158-X SLMO line cards

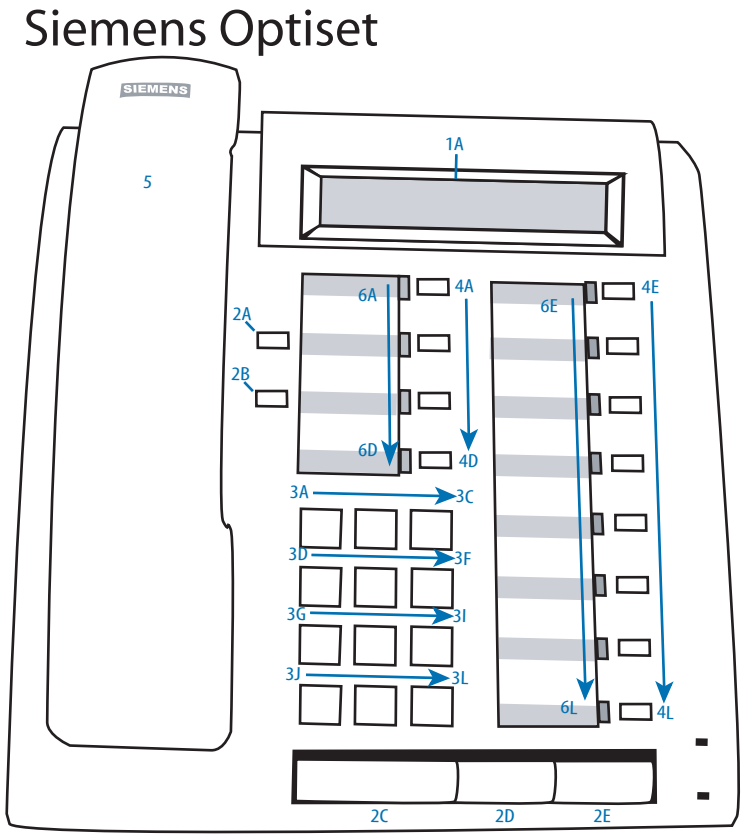
HiPath 3000 & 3750 - HE580S.04.517 Version 4 and Q2901-X SLMO line card

HiPath 4000 - SMR10 SMP4 Version 1.0 and Q2168-X SLMO line card

OPTISET EADVANCE OR EADVANCED CONFERENCE

PHONE MAP

The following events were observed when each phone button was used.



1A
EVT_MESSAGE_CHANGE

2A
EVT_SOFT_BUTTON_PRESSED
Subreason = 0x00000000
2B
EVT_SOFT_BUTTON_PRESSED
Subreason = 0x00000001
2C
EVT_SOFT_BUTTON_PRESSED
Subreason = 0x00000004
2D
EVT_SOFT_BUTTON_PRESSED
Subreason = 0x00000002
2E
EVT_SOFT_BUTTON_PRESSED
Subreason = 0x00000003

3A–3L
EVT_DIGIT_PRESSED
Subreasons: Correspond to digit pressed

4A–4L
EVT_FUNCTION_BUTTON_PRESSED
Subreasons: 0x00000000–0x00000011

5
EVT_OFF_HOOK, EVT_ON_HOOK

6A-6L
EVT_FUNCTION_LIGHT_ON
EVT_FUNCTION_LIGHT_OFF
EVT_FUNCTION_LIGHT_FASTFLASHING
EVT_FUNCTION_LIGHT_FLASHING
Subreason: 0x00000000–0x00000011

Chapter 12

Troubleshooting

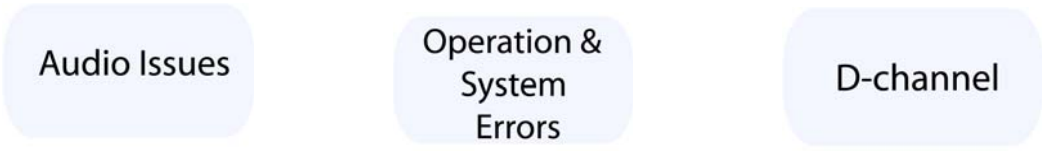
This section provides troubleshooting tips for common problems when using the SmartWORKS NGX.

This chapter has been designed much like a web-site. Links are provided for navigation.

NGX Troubleshooting

The NGX troubleshooting section has been broken down into the following categories.

Select a topic to begin:



Back to NGX Main

Audio Issues

SELECT A PROBLEM:

Solution

ONE CHANNEL SEEMS BAD

1. Verify that the phone models and PBX are supported by the NGX. This information is posted on Online Help > Files & Documents > NGX Matrix.
2. In SmartVIEW under Framer Statistics for NGX -verify that there is Amplitude on the line and there are no TE, NT or sync errors on bad channel. NT value should also be greater than TE value. Errors usually exist for the following reason:
   a) PBX or phone is not supported (or phone add-on modules)
   b) Verify cable lengths do not exceed AudioCodes recommendations. This information is posted on Online Help > Files & Documents > NGX Matrix.
   c) Check cable connections and wiring of recording system, correct any loose cables
   d) Once cables are corrected, retest.
3. Validate channel by swapping lines with a known good channel.
4. Check the Windows System Event Viewer for NtiDrv errors/warnings and contact tech support is any are present.
5. Contact Tech support if above solutions do not resolve issue. When contacting tech support, please provide the following:
   - SmartWORKS version
   - cable lengths (tap to phones, tap to NGX, PBX to tap)
   - PBX model
   - phone models
   - Framer Statistics - errors and Amplitude readings
   - Note errors in Windows System Event Log. Send .evt file to AudioCodes Tech support for evaluation (compressed .zip file preferred)

Top

Solution

DISTANT PHONE SOUNDS WEAK

NOTE: As of SmartWORKS release 3.2, new APIs have been added that provide separate gain control on primary and secondary inputs.

1. Verify that AGC (Automatic Gain Control() is enabled.
2. Verify the correct AGC settings (this is used to equalize the sound on both ends). This can be tested by using SmartVIEW. Go to Settings>MT(Get/Set)AGC. (if using v. 3.2 or later you can use MTChInputSetAGC). Set Target Max Amplitude to 12-15 dBm and set Max Amplification to -10 dBm. Leave Attack and Decay values as default. Then check Enable, click Set and click Close. Make a test call to verify sound quality. If the volume of both sides of the conversation are the same continue by setting Volume. Use MTSetGain() (if using v3.2 or late use MTChInputSetGain()) to adjust the volume.

3. In SmartVIEW under Framer Statistics for NGX -verify that there is Amplitude on the line and there are no TE, NT or sync errors on bad channel. Also NT values should be greater than TE values. If errors exist then:
   a) PBX or phone is not supported (or phone add-on modules)
   b) Verify cable lengths do not exceed AudioCodes recommendations. This information is posted on Online Help > Files & Documents > NGX Matrix.
   c) Check cable connections and wiring of recording system, correct any loose cables
   d) Once cables are corrected, retest.
4. Some PBX have AGC built in. You may need to adjust the settings.
5. Contact Tech support if above solutions do not resolve issue. When contacting tech support, please provide the following:
   - Provide a sample recording, using SmartVIEW use MTRecFile() and record file in A-law 16-bit linear format
   - SmartWORKS version
   - cable lengths (tap to phones, tap to NGX, PBX to tap)
   - PBX model
   - phone models
   - Framer Statistics - errors and Amplitude readings
   - Note errors in Windows System Event Log. Send .evt file to AudioCodes Tech support for evaluation (compressed .zip file preferred)

Top

Solution

POOR RECORDING QUALITY

1. Verify proper audio format - make sure recording format matches playback format.
2. Verify A-Law\µ-Law settings. Generally speaking, the following holds true: µ-Law is used in North America and Japan, A-Law is used in Europe. This can also vary with PBX model, check PBX documentation.
3. In SmartVIEW under Framer Statistics for NGX -verify that there is Amplitude on the line and there are no TE, NT or sync errors on bad channel. If errors exist then:
   a) PBX or phone is not supported (or phone add-on modules)
   b) Verify cable lengths do not exceed AudioCodes recommendations. This information is posted on Online Help > Files & Documents > NGX Matrix.
   c) Check cable connections and wiring of recording system, correct any loose cables
   d) Once cables are corrected, retest.
4. Ensure the headset is of good quality.

~ more on next page~

5. Generate a test recording using SmartVIEW (on the navigation menu go to Media > ***MTRecFile()***). Record test file in µ-Law or A-law 16 bit linear with AGC DISABLED.

6. Contact Tech support if above solutions do not resolve issue. When contacting tech support, please provide the following:
   - Provide a sample recording, using SmartVIEW use MTRecFile() and record file in A-law 16-bit linear format
   - SmartWORKS version
   - cable lengths (tap to phones, tap to NGX, PBX to tap)
   - PBX model
   - phone models
   - Framer Statistics - errors and Amplitude readings
   - Note errors in Windows System Event Log. Send .evt file to AudioCodes Tech support for evaluation (compressed .zip file preferred)

Top

Solution

NOTABLE GAPS IN RECORDING

1. In SmartVIEW under Framer Statistics for NGX -verify that there is Amplitude on the line and there are no TE, NT or sync errors on bad channel. If errors exist then:
   a) PBX or phone is not supported (or phone add-on modules)
   b) Verify cable lengths do not exceed AudioCodes recommendations. This information is posted on Online Help > Files & Documents > NGX Matrix.
   c) Check cable connections and wiring of recording system, correct any loose cables
   d) Once cables are corrected, retest.
2. If your application is using streaming or using ***MTRecBuffer()***, verify no buffers are being dropped. To verify this, review the Windows Event Viewer for Encode Queue overflow errors. **NOTE:** When using Linux, all information is written to a 'messages' file located in the /var/log directory.
3. Check for gaps by listening to a sample recording. Using SmartVIEW use MTRec-File() and record in A-law 16-bit linear format.
4. Verify recording using SmartVIEW Test Application, send to tech support if the sample is bad.
5. Verify Mixing is not disabled.
6. Contact Tech support if above solutions do not resolve issue. When contacting tech support, please provide the following:
   - SmartWORKS version
   - cable lengths (tap to phones, tap to NGX, PBX to tap)
   - PBX model
   - phone models

- Framer Statistics - errors and Amplitude readings
- Note errors in Windows System Event Log. Send .evt file to AudioCodes Tech support for evaluation (compressed .zip file preferred)

Top

Solution

UNABLE TO RECORD ONE SIDE OF A CONVERSATION

1. Verify that mixing has not been disabled.
2. If using a secondary DSP card to record, ensure ***MTSetOutputSource()*** for channel is set to AUDIO_STREAM.
3. Check for gaps by listening to a sample recording. Using SmartVIEW use MTRecFile() and record in A-law 16-bit linear format.
4. Contact Tech support if above solutions do not resolve issue. When contacting tech support, please provide the following:
   - SmartWORKS version
   - cable lengths (tap to phones, tap to NGX, PBX to tap)
   - PBX model
   - phone models
   - Framer Statistics - errors and Amplitude readings
   - Note errors in Windows System Event Log. Send .evt file to AudioCodes Tech support for evaluation (compressed .zip file preferred)

Top

Solution

DAUGHTER CARD CHANNELS DO NOT WORK, BUT BASED CARD CHANNELS DO

1. Confirm that phone models are supported by the NGX. This information is posted in Online Help > Files & Documents > NGX Matrix.
2. When changing PBX's in Smart Control, ensure Apply button is pressed for Base and daughter cards individually, then restart driver and\or system.
3. If only one daughter card is installed, make sure it is installed in the position that is closest to the bracket with the RJ21 connector.
4. Verify screws holding daughter card are securely fastened.
5. In SmartVIEW under Framer Statistics for NGX -verify that there is Amplitude on the line and there are no TE, NT or sync errors on bad channel. Also - NT value should be greater than TE value. Errors usually exist for the following reasons:
   a) PBX or phone is not supported (or phone add-on modules)
   b) Verify cable lengths do not exceed AudioCodes recommendations. This information is posted on Online Help > Files & Documents > NGX Matrix.
   c) Check cable connections and wiring of recording system, correct any loose cables
   d) Once cables are corrected, retest.

~ more on next page~

6. Once cables are corrected, retest.
7. Check Event Viewer for NtiDrv errors/warnings.
8. Contact Tech support if above solutions do not resolve issue. When contacting tech support, please provide the following:
   - SmartWORKS version
   - cable lengths (tap to phones, tap to NGX, PBX to tap)
   - PBX model
   - phone models
   - Framer Statistics - errors and Amplitude readings
   - Note errors in Windows System Event Log. Send .evt file to AudioCodes Tech support for evaluation (compressed .zip file preferred)

Top

Back to NGX Main

Operation and System Errors

SELECT A PROBLEM:

Solution

BOARD PANICS

1. Verify that board is seated properly in the host PC.
2. If using SmartWORKS v 2.9.x or earlier:

   a) Hyper threading and DUAL CPU is not supported, upgrade to 2.10 or higher
3. Enable trace log and send the log to Tech. Support. (To enable trace log, review the application note: *Enabling Trace Log*).
4. Send Windows System event log (.evt format) zipped to AudioCodes support for review.

Top

Solution

LED INDICATIONS

1. CR1--- DSP started LED, if the LED is on, the board downloaded successfully. This doesn't occur until the first time the application starts.
   On daughter card CR1 may be on\off before loading.
2. CR2---
   "Green" indicates all channels
   synchronized with PBX
   "Amber" indicates one or more channels are not synchronized
   "Red" indicates No channels are synchronized.
3. CR7--- "ON" indicates the boards TDM (MVIP,H100) clocks are being terminated.
4. CR17---
   "ON" indicates the board download is successful
   "Off" indicates no communication with board
   "Blinking" indicates driver has started but no download has occurred yet
5. On daughter card, there is DSP LED that is "on" when daughter card is successfully downloaded.
6. Contact Tech support if above solutions do not resolve issue. When contacting tech support, please provide the following:
   - SmartWORKS version
   - cable lengths (tap to phones, tap to NGX, PBX to tap)
   - PBX model
   - phone models
   - Framer Statistics - errors and Amplitude readings
   - Note errors in Windows System Event Log. Send .evt file to AudioCodes Tech support for evaluation (compressed .zip file preferred)

Top

Solution

SYCN. ERRORS (TE/NT)

1. Validate that the phone and PBX model and line card is supported. This information is available on Online Help > Files & Documents > NGX PBX Matrix.
2. Verify cable lengths do not exceed AudioCodes recommendations. This information is available on Online Help > Files&Documents > NGX PBX Matrix.
3. Verify proper PBX image loaded into NGX and that the driver was restarted correctly after load.
4. Verify no intermittent connections on wiring.
5. Verify no other devices are parallel tapping the line.
6. Verify termination in Smart Control Panel. Standard default is Hi-impedance. 120 ohms termination is only for special cases.
7. If the cable length from PBX tap to the phone is greater than the tap to the phones, try increasing the tap to phone length.

Top

Solution

MVIP/H.100 TIMESLOT ISSUE

1. Ensure selection between MVIP and H.100 is correct in Control Panel. (For H.100 ensure correct bus speed.)
2. Ensure MVIP or H.100 cable is correctly attached between boards.
3. Ensure proper termination of MVIP or H.100 bus if more than 5 boards are used. Look in Developer's manual for more information.
4. Ensure only one Master is selected for MVIP, and Master A and/or Master B for H.100 Interface.

~ con't on next page

5. Ensure "Allow bus segmentation" is not checked in Smart Control Panel when using MVIP or H.100 bus. Remember to restart drivers is any changes are made in the Control Panel.
6. Contact Tech support if above solutions do not resolve issue. When contacting tech support, please provide the following:
   - SmartWORKS version
   - cable lengths (tap to phones, tap to NGX, PBX to tap)
   - PBX model
   - phone models
   - Framer Statistics - errors and Amplitude readings
   - Note errors in Windows System Event Log. Send .evt file to AudioCodes Tech support for evaluation (compressed .zip file preferred)

Top

Solution

LOADING PBX IMAGES

1. Ensure that when changing PBX types in Smart Control Panel that each daughter card and the base card are changed by selecting "Apply" for each section.
2. Ensure that driver is restarted or system reset to load in new PBX types after performing above.
3. Contact Tech support if above solutions do not resolve issue. When contacting tech support, please provide the following:
   - SmartWORKS version
   - cable lengths (tap to phones, tap to NGX, PBX to tap)
   - PBX model
   - phone models
   - Framer Statistics - errors and Amplitude readings
   - Note errors in Windows System Event Log. Send .evt file to AudioCodes Tech support for evaluation (compressed .zip file preferred)

Top

Solution

BOARD INSTALL PROBLEM

1. Verify PC is 2.2 PCI with 3.3v compliant.
2. Ensure reboot after initial installation of software package.
3. Verify board fits properly in PCI slot.
4. Check in Device Manager to see if the board is recognized.

~ more on next page ~

5. Ensure Power LED is on-refer to LED section.
6. Verify PC Power Supply meets power requirements of all devices installed including SmartWORKS board.
7. If another version of SmartWORKS was previously installed, it may not have been un-installed correctly. Un-install SmartWORKS - reboot, re-install Smart-WORKS - reboot.
8. Verify the current version of SmartWORKS supports your computer's operating system.

Windows NT	v2.10 or earlier (SP 6 is required)
Windows 2000	All versions. (SP 3 required)
Windows 2003 32 bit	v 3.0 or later (
Windows XP	v. 2.10 or later (SP 1 is required)
Linux	Contact support for availability

NOTE: When installing over Windows NT or 2000, users must cancel out of "New Hardware Found" wizard. When installing over Windows XP or 2003 the operating system must load the driver so do NOT cancel out of the "New Hardware Found" wizard.

9. Contact Tech support if above solutions do not resolve issue. When contacting tech support, please provide the following:
   - SmartWORKS version
   - cable lengths (tap to phones, tap to NGX, PBX to tap)
   - PBX model
   - phone models
   - Framer Statistics - errors and Amplitude readings
   - Note errors in Windows System Event Log. Send .evt file to AudioCodes Tech support for evaluation (compressed .zip file preferred)

Top

Solution

ON BOARD AUDIO JACK IS NOT WORKING

1. Verify proper wiring to audio connector-see User's manual for details.
2. Verify that the correct settings are used when routing media from on-board DSP resources to the audio jack. Complete instructions, along with diagrams, are available in the Function Reference Library. Refer to the page that defines the MTSetOutputSource() function. This page provides step by step instructions for using ***MTSetOutputSource()***, ***MTSetOutput()***, and ***MTAJListen()***.
3. If multiple boards are in the system, verify that the correct board number has been selected.
4. Contact Tech support if above solutions do not resolve issue. When contacting tech support, please provide the following:
   - SmartWORKS version
   - cable lengths (tap to phones, tap to NGX, PBX to tap)
   - PBX model
   - phone models
   - Framer Statistics - errors and Amplitude readings (not required for this issue)
   - Note errors in Windows System Event Log. Send .evt file to AudioCodes Tech support for evaluation (compressed .zip file preferred)

Top

Back to NGX Main

D-channel Problems

SELECT A PROBLEM:

Solution

EVT_MESSAGE_CHANGE DATA AND PHONE DISPLAY DO NOT MATCH

1. Verify phone and PBX model is supported in NGX PBX Support list, if not then contact Sales department to discuss business case - bladesinfo@audio-codes.com.
2. Use SmartVIEW to verify EVT_MESSAGE_CHANGE event. The display on the phone may change several times during the course of the call. Your application may be storing the wrong display message
3. In SmartVIEW under Framer Statistics for NGX -verify that there is Amplitude on the line and there are no TE, NT or sync errors on bad channel. NT value should be greater than TE value. Errors usually exist for the following reason:
   a) PBX or phone is not supported (or phone add-on modules)
   b) Verify cable lengths do not exceed AudioCodes recommendations. This information is posted on Online Help > Files & Documents > NGX Matrix.
   c) Check cable connections and wiring of recording system, correct any loose cables
   d) Once cables are corrected, retest.
4. Request copy of SmartCapture from Support which will allow you to capture raw D-Channel on site. Then return the output file to AudioCodes for review.

Top

Solution

DUPLICATE EVENTS

1. Disable Event updates in Control Panel.
2. Monitor D-channel events in SmartVIEW and verify duplicate events are visible.
3. Request copy of SmartCapture from Support which will allow you to capture raw D-Channel on site. Then return the output file to AudioCodes for review.
4. Enable trace log and send the log to Tech. Support. (To enable trace log, review the application note: *Enabling Trace Log*).

Top

Solution

NO D-CHANNEL EVENTS

1. Verify the PBX and phone model is supported for D-Channel. This information is available in Online Help>Files & Documents>NGX PBX Matrix. If it is not supported, contact sales to discuss the business case - bladesinfo@audiocodes.com.
2. Verify that D-Channel is checked in Smart Control for the base board and each daughter card. If changes are made, drivers must be restarted.
3. In order to receive events in your application the board and channel must be opened as all D-channel events are board events. AudioCodes recommends using MTSysStartup to open the board and all channels.
4. In SmartVIEW under Framer Statistics for NGX -verify that there is Amplitude on the line and there are no TE, NT or sync errors on bad channel. NT value should also be greater than TE value. If errors exist then:
   a) PBX or phone is not supported (or phone add-on modules)
   b) Verify cable lengths do not exceed AudioCodes recommendations. This information is posted on Online Help > Files & Documents > NGX Matrix.
   c) Check cable connections and wiring of recording system, correct any loose cables
   d) Once cables are corrected, retest.
5. Verify there are no CRC errors in the Windows System event log. CRC errors are indicative of a cabling problem or some new condition, line card or phone that has not be tested.
6. Contact Tech support if above solutions do not resolve issue. When contacting tech support, please provide the following:
   - SmartWORKS version
   - cable lengths (tap to phones, tap to NGX, PBX to tap)
   - PBX model
   - phone models
   - Framer Statistics - errors and Amplitude readings
   - Note errors in Windows System Event Log. Send .evt file to AudioCodes Tech support for evaluation (compressed .zip file preferred)

Top

Appendix A

Event Sequences

Introduction

The following section outlines event sequences generated between phones that are tapped by a SmartWORKS board on a supported PBX. Where possible, phone illustrations have been included that display how phone buttons corollate to various events in the SmartWORKS API.

This information will be incorporated into this document as more chapters are included that detail how to integrate the NGX with specified PBXs.

Avaya Merlin Magix Event Sequences

Following is a list of events that can be expected using the NGX with an Avaya Merlin Magix PBX. The recorded D-channel events have also been provided on CD to demonstrate the sequence of events.

EXAMPLE SEQUENCE

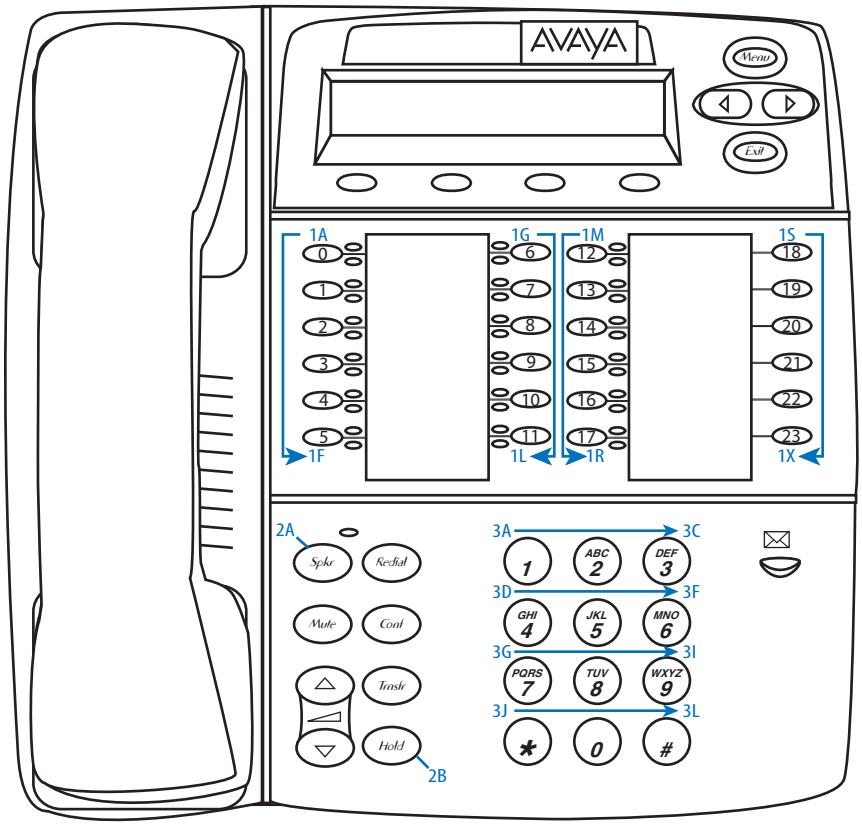
A Call from an extension connected to Channel 3 to an extension connected to Channel 2.

```
CHANNEL      EVENT                          CAUSE
Channel 3    EVT_AUDIO_ON(Handset 1)        // phone (X102) picked up
Channel 3    EVT_OFFHOOK
Channel 3    EVT_CONNECTED(0x0000)
Channel 3    EVT_LIGHT_ON(GREEN 0)
Channel 3    EVT_LIGHT_ON(RED 0)
Channel 3    EVT_MESSAGE_CHANGE (17)
Channel 3
Channel 3    EVT_DIGIT_PRESSED(1)           // phone (X102) dialed "1"
Channel 3    EVT_DIGIT_RELEASED(1)
Channel 3    EVT_MESSAGE_CHANGE (17)
Channel 3    1
Channel 3    EVT_DIGIT_PRESSED(0)           // phone (X102) dialed "0"
Channel 3    EVT_DIGIT_RELEASED(0)
Channel 3    EVT_MESSAGE_CHANGE (17)
Channel 3    10
Channel 3    EVT_DIGIT_PRESSED(1)           // phone (X102) dialed "1"
Channel 3    EVT_DIGIT_RELEASED(1)
Channel 3    EVT_CONNECTED(0x0000)
Channel 2    EVT_RING_ON(1)
Channel 2    EVT_LIGHT_OFF(GREEN 0)
Channel 2    EVT_LIGHT_OFF(RED 0)
Channel 2    EVT_LIGHT_FLASHING(GREEN 1)
Channel 2    EVT_LIGHT_OFF(RED 1)
Channel 2    EVT_LIGHT_ON(RED 1)
Channel 2    EVT_MESSAGE_CHANGE (25)
Channel 2    JOE     Ext102
Channel 3    EVT_MESSAGE_CHANGE (17)
Channel 3    PETER   Ext101
Channel 2    EVT_AUDIO_ON(Handset 1)        // phone (X101) picked up
Channel 2    EVT_OFFHOOK
Channel 2    EVT_RING_OFF(1)
Channel 2    EVT_CONNECTED(0x0000)
Channel 2    EVT_LIGHT_ON(GREEN 1)
Channel 2    EVT_MESSAGE_CHANGE (25)
Channel 2    JOE     Ext102
Channel 3    EVT_AUDIO_OFF(Handset 1)       // phone (X102) hung up
Channel 3    EVT_ONHOOK
Channel 3    EVT_DISCONNECTED(0x0000)
Channel 2    EVT_DISCONNECTED(0x0000)
Channel 3    EVT_LIGHT_OFF(GREEN 0)
Channel 3    EVT_LIGHT_OFF(RED 0)
Channel 2    EVT_LIGHT_OFF(GREEN 1)
Channel 2    EVT_LIGHT_OFF(RED 1)
Channel 2    EVT_DISCONNECTED(0x0000)
Channel 3    EVT_LIGHT_ON(RED 0)
Channel 2    EVT_MESSAGE_CHANGE (25)
Channel 2
```

{Continues on next page}

CHANNEL	EVENT	CAUSE
Channel 3	EVT_MESSAGE_CHANGE (17)	
Channel 3		
Channel 2	EVT_AUDIO_OFF(Handset 1)	// phone (X102) hung up
Channel 2	EVT_ONHOOK	
Channel 2	EVT_LIGHT_ON(RED 0)	
Channel 2	EVT_MESSAGE_CHANGE (25)	
Channel 2		

PHONE MAP



1A–X
EVT_FUNCTION_BUTTON_PRESSED
Subreasons:
0x00000000– 0x00000017

2A
EVT_SPEAKER_BUTTON_PRESSED
EVT_SPEAKER_BUTTON_RELEASED
2B
EVT_HOLD_BUTTON_PRESSED
EVT_HOLD_BUTTON_RELEASED

3A–3L
EVT_DIGIT_PRESSED
EVT_DIGIT_RELEASED
Subreasons: Correspond to digit pressed or released

Ericsson Event Sequences

Following is a list of events that can be expected using the NGX with an Ericsson MD110 (D-channel) PBX. The recorded D-channel events have also been provided on CD to demonstrate the sequence of events.

Application Development Notes

As of release 3.5.0, the SmartWORKS DLL supports both the ELU25 and ELU28 line cards. This release also supports Ericsson's DIALOG 4220, 4222, 4223, and 4225 phones. Both line cards, and all phones are supported with the same NGX firmware. All AudioCodes regressions for the ELU25 and phones passed with only one concern issued, namely the first phone and PBX message issued after the firmware is started is lost to guarantee that no partial messages are processed.

The following behaviors have been noted by AudioCodes and should be noted by application developers:

- Speaker audio and handset audio control messages sent by the PBX vary from the behavior seen on previous Ericsson phones, therefore existing applications may be affected. Application developers are encouraged to test this behavior prior to porting applications to newer phones using the ELU28 line card.
- ONHOOK / AUDIO_CHANGE events - On-hook messages are not sent by the DIALOG 4222, 4223 and 4225 phones when the speaker is active and the handset is put on hook. For example, when an agent answers the phone using the handset an EVT_OFFHOOK event is reported, along with EVT_AUDIO_CHANGE indicating the handset's audio has been enabled. Should the agent put this call on speaker and replace the handset, the phone does not report an on-hook condition to the PBX therefore the EVT_ONHOOK is not reported. These phones also do not require an audio off command from the PBX to disable the audio on the handset. When the agent releases the call by pressing the speaker phone button, an audio off command is passed to the phone in regards to the speaker audio. At this point the EVT_AUDIO_CHANGE is reported to the user application indicating that the audio has been disabled. **NOTE:** When tapping DIALOG 4220 phones, the EVT_AUDIO_CHANGE is reported when the handset is replaced indicating that the handset's audio has been disabled.
- EVT_MESSAGE_CHANGE - it has been observed that a single phone, when connected to different line cards, require a different message length value. When the EVT_MESSAGE_CHANGE is reported to the user application by the NGX, the length field of the MT_EVENT data structure informs the user of the proper message length.
- Stored Messages - DIALOG 4225 phones, when connecting to the network, receive a list of messages from the PBX which are stored on the phone. The phone manages the display of these messages without direction from the PBX. As a result, when using this phone model, the application does not receive an EVT_MESSAGE_CHANGE with each change to the phone's LCD.

Example Sequence

A Call from an extension connected to Channel 3 to an extension connected to Channel 7.

```
CHANNEL      EVENT                          CAUSE
Channel 3      EVT_OFFHOOK                  //Phone (X2002) picked up
Channel 3      EVT_LIGHT_ON(RED 11)
Channel 3      EVT_AUDIO_ON(Handset 1)
Channel 3      EVT_MESSAGE_CHANGE (84)
Channel 3
Channel 3
Channel 3
Channel 3    LOCK ACC
Channel 3      EVT_MESSAGE_CHANGE (84)
Channel 3
Channel 3
Channel 3
Channel 3    LOCK ACC AUTH TIMER
Channel 3      EVT_DIGIT_PRESSED(2)         //Phone (X2002) dialed "2"
Channel 3      EVT_MESSAGE_CHANGE (84)
Channel 3
Channel 3
Channel 3                     2
Channel 3      EVT_DIGIT_PRESSED(0)         //Phone (X2002) dialed "0"
Channel 3      EVT_MESSAGE_CHANGE (84)
Channel 3
Channel 3
Channel 3                    20
Channel 3      EVT_DIGIT_PRESSED(0)         //Phone (X2002) dialed "0"
Channel 3      EVT_MESSAGE_CHANGE (84)
Channel 3
Channel 3
Channel 3                   200
Channel 3      EVT_DIGIT_PRESSED(6)         //Phone (X2002) dialed "6"
Channel 3      EVT_MESSAGE_CHANGE (84)
Channel 3
Channel 3
Channel 3                  2006
Channel 7      EVT_LIGHT_FASTFLASHING(RED 11)
Channel 3      EVT_MESSAGE_CHANGE (84)
Channel 3
Channel 3    CALLBACK =
Channel 3      EVT_MESSAGE_CHANGE (84)
Channel 3
Channel 3    CALLBACK = 6
Channel 7      EVT_RING_ON(1)
Channel 3      EVT_MESSAGE_CHANGE (84)
Channel 3
Channel 3    CALLBACK = 6
Channel 3                  2006
Channel 3     CAB  PAG
Channel 3      EVT_MESSAGE_CHANGE (84)
```

{Continues on next page}

```
CHANNEL      EVENT                          CAUSE
Channel 3
Channel 3    CALLBACK = 6
Channel 3                  2006
Channel 3     CAB  PAG      TIMER
Channel 7       EVT_MESSAGE_CHANGE (84)
Channel 7             J VanPelt
Channel 7      EVT_MESSAGE_CHANGE (85)
Channel 7    2002
Channel 7      EVT_OFFHOOK                  //Phone (X2006) picked up
Channel 7      EVT_AUDIO_ON(Handset 1)
Channel 3      EVT_MESSAGE_CHANGE (84)
```

```
Channel 3  TIME & DATE NOT SET
Channel 3     EVT_MESSAGE_CHANGE (84)
Channel 3  TIME & DATE NOT SET
Channel 3
Channel 3                  2006
Channel 3   ETE  ACC
Channel 7     EVT_MESSAGE_CHANGE (84)
Channel 7  2002       J VanPelt
Channel 7                  2002
Channel 3     EVT_MESSAGE_CHANGE (84)
Channel 3  TIME & DATE NOT SET
Channel 3
Channel 3                  2006
Channel 3   ETE  ACC      TIMER
Channel 7     EVT_MESSAGE_CHANGE (84)
Channel 7  2002       J VanPelt
Channel 7  J VanPelt
Channel 3     EVT_MESSAGE_CHANGE (84)
Channel 3  TIME & DATE NOT SET
Channel 3
Channel 3                  2006
Channel 7     EVT_MESSAGE_CHANGE (84)
Channel 7  TIME & DATE NOT SET
Channel 7     EVT_RING_OFF(1)
Channel 7     EVT_LIGHT_ON(RED 11)
Channel 3     EVT_ONHOOK                 //Phone (X2002) hung up
Channel 3     EVT_AUDIO_OFF(Handset 1)
Channel 7     EVT_AUDIO_OFF(Handset 1)
Channel 7     EVT_MESSAGE_CHANGE (85)
Channel 7  TIME & DATE NOT SET
Channel 7
Channel 7     EVT_LIGHT_OFF(RED 11)
Channel 3     EVT_MESSAGE_CHANGE (84)
Channel 3  TIME & DATE NOT SET
Channel 3     EVT_LIGHT_OFF(RED 11)
Channel 7     EVT_MESSAGE_CHANGE (84)
Channel 7  TIME & DATE NOT SET
Channel 3     EVT_MESSAGE_CHANGE (84)
Channel 3  TIME & DATE NOT SET
Channel 3
Channel 3  J VanPelt
Channel 7     EVT_MESSAGE_CHANGE (84)
Channel 7  TIME & DATE NOT SET
Channel 7
Channel 3     EVT_MESSAGE_CHANGE (84)
Channel 3  TIME & DATE NOT SET
Channel 3
Channel 3  J VanPelt       2002
```

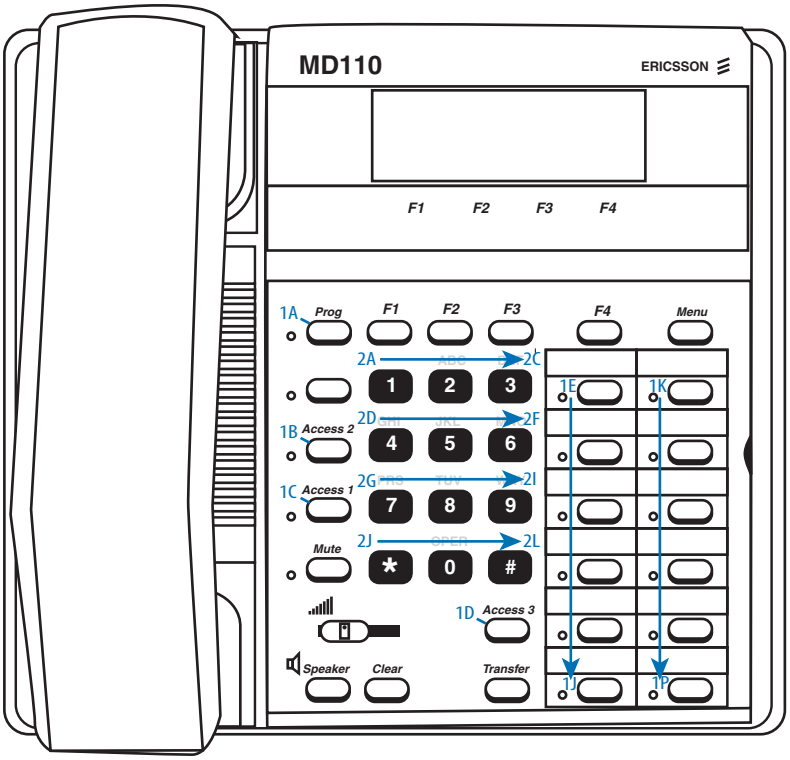
{Continues on next page}

```
CHANNEL      EVENT                      CAUSE
Channel 7     EVT_MESSAGE_CHANGE (84)
Channel 7  TIME & DATE NOT SET
Channel 7                  2006
Channel 3     EVT_MESSAGE_CHANGE (84)
Channel 3  TIME & DATE NOT SET
Channel 3     EVT_MESSAGE_CHANGE (84)
Channel 3  TIME & DATE NOT SET
Channel 3
Channel 3  J VanPelt       2002
Channel 3
Channel 3     EVT_MESSAGE_CHANGE (84)
```

```
Channel 3  TIME & DATE NOT SET
Channel 3
Channel 3  J VanPelt       2002
Channel 3
Channel 7     EVT_ONHOOK                  //Phone (X2006) hung up
```

PHONE MAP

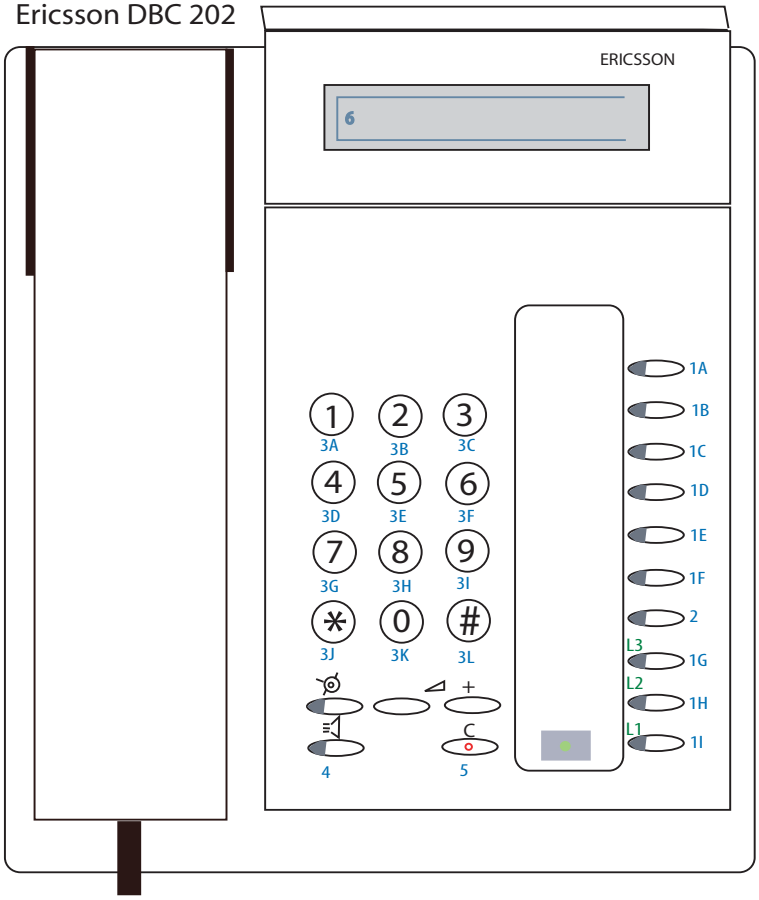
Ericsson DBC662



1A–1P
EVT_FUNCTION_BUTTON_PRESSED
1A
Subreason: 0x00000000
1B
Subreason: 0x00000010
1C
Subreason: 0x00000011
1D
Subreason: 0x00000009
1E
Subreasons: 0x00000014–0x00000025

2A–2L
EVT_DIGIT_PRESSED
Subreasons: Correspond to digit pressed

Ericsson DBC 202



1A–1I
EVT_FUNCTION_BUTTON_PRESSED
Subreasons:
1A - 0x00000000
1B - 0x00000001
1C - 0x00000002
1D - 0x00000003
1E - 0x0000000E
1F - 0x0000000D
1G - 0x00000009
1H- 0x0000000A
1I - 0x0000000B

2 EVT_TRANSFER_BUTTON_PRESSED

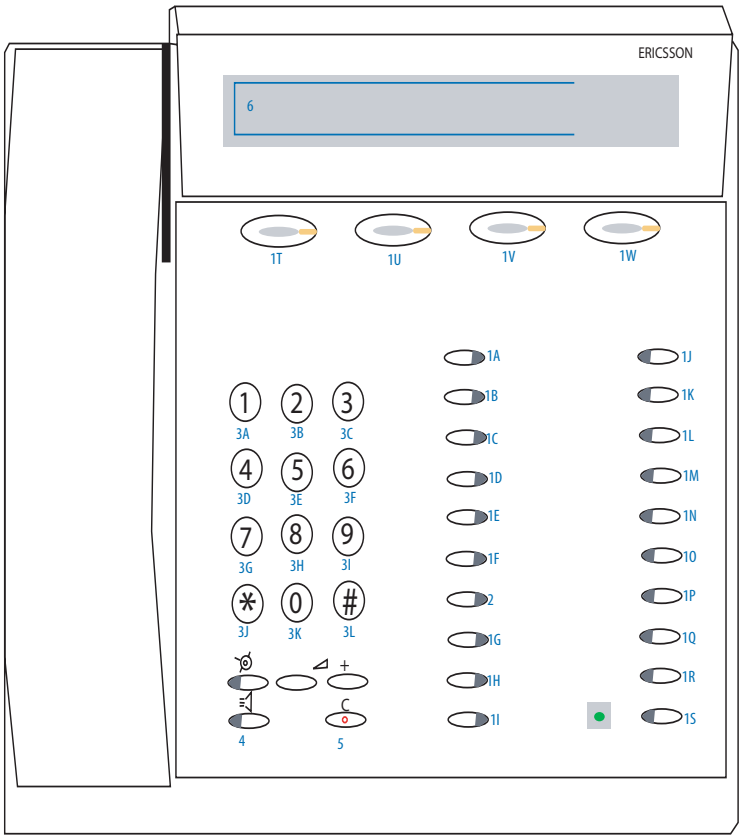
3A–3L
EVT_DIGIT_PRESSED
Subreasons: Correspond to digit pressed
* - 0x0000002A
# - 0x00000023

4
EVT_SPEAKER_BUTTON_PRESSED
Subreason: 0x00000000
5
EVT_RELEASE_BUTTON_PRESSED
Subreason: 0x00000000
6
EVT_MESSAGE_CHANGE
Subreason: 0x00000000

L1-L3
EVT_LIGHT_ON
EVT_LIGHT_OFF
EVT_LIGHT_QUICKFLASH

L1
L2
L3

Ericsson DBC203



1A–1I
EVT_FUNCTION_BUTTON_PRESSED
Subreasons:
1A - 0x00000000
1B - 0x00000001
1C - 0x00000002
1D - 0x00000003
1E - 0x0000000E
1F - 0x0000000D
1G - 0x00000009
1H- 0x0000000A
1I - 0x0000000B
1J - 0x00000008
1K - 0x0000000F
1L - 0x00000010
1M - 0x00000011
1N - 0x00000012
1O - 0x00000013
1P - 0x00000014
1Q- 0x00000015
1R - 0x00000016
1S - 0x00000017

1T- 0x00000004
1U - 0x00000005
1V - 0x00000006
1W - 0x00000007

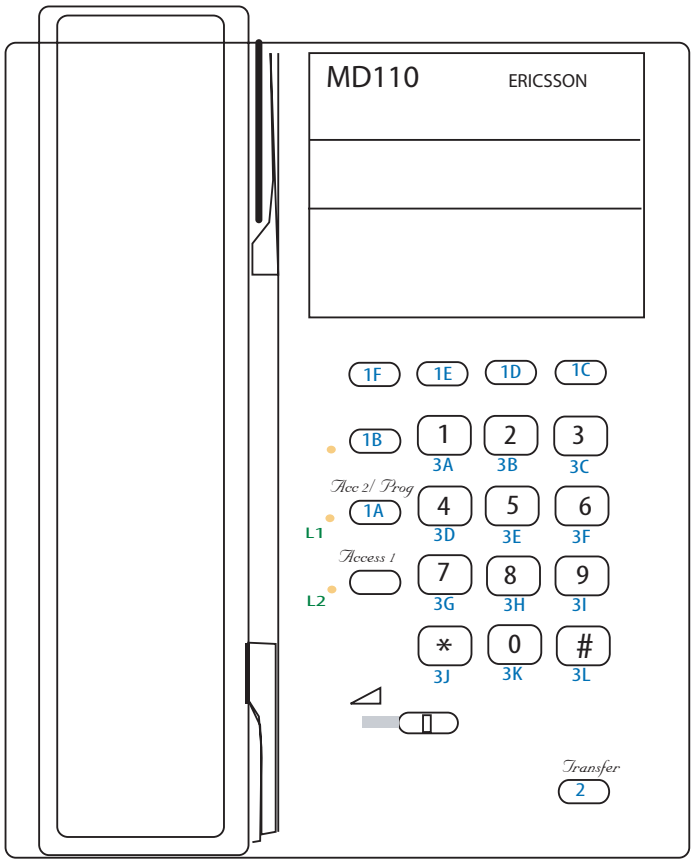
2 EVT_TRANSFER_BUTTON_PRESSED

3A–3L
EVT_DIGIT_PRESSED
Subreasons: Correspond to digit pressed
* - 0x0000002A
# - 0x00000023

4
EVT_SPEAKER_BUTTON_PRESSED
Subreason: 0x00000000
5
EVT_RELEASE_BUTTON_PRESSED
Subreason: 0x00000000
6
EVT_MESSAGE_CHANGE
Subreason: 0x00000000

Light events have not been documented yet.

Ericsson DBC501



1A–1F
EVT_FUNCTION_BUTTON_PRESSED
Subreasons:

1A - 0X0000000A
1B - 0X00000002
1C - 0X00000004
1D - 0X00000005
1E - 0X00000006
1F - 0X00000007

2
EVT_TRANSFER_BUTTON_PRESSED

3
EVT_DIGIT_BUTTON_PRESSED
Subreasons:

3A - 0X00000031
3B - 0X00000032
3C - 0X00000033
3D - 0X00000034
3E - 0X00000035
3F - 0X00000036
3G - 0X00000037
3H - 0X00000038
3I - 0X00000039
3J - 0X0000002A
3K - 0X00000030
3L - 0X00000023

L1
EVT_LIGHT_ON
EVT_LIGHT_OFF
EVT_LIGHT_QUICKFLASH
EVT_LIGHT_FLASH
Subreasons:

L2
EVT_LIGHT_ON
EVT_LIGHT_OFF
EVT_LIGHT_QUICKFLASH
EVT_LIGHT_FLASH
Subreasons:

NEC Electra Elite Event Sequences

Following is a list of events that can be expected using the NGX with an NEC Electra Elite on the Nortel DTU-32D-2. The recorded D-channel events have also been provided on CD to demonstrate the sequence of events.

Example Sequence

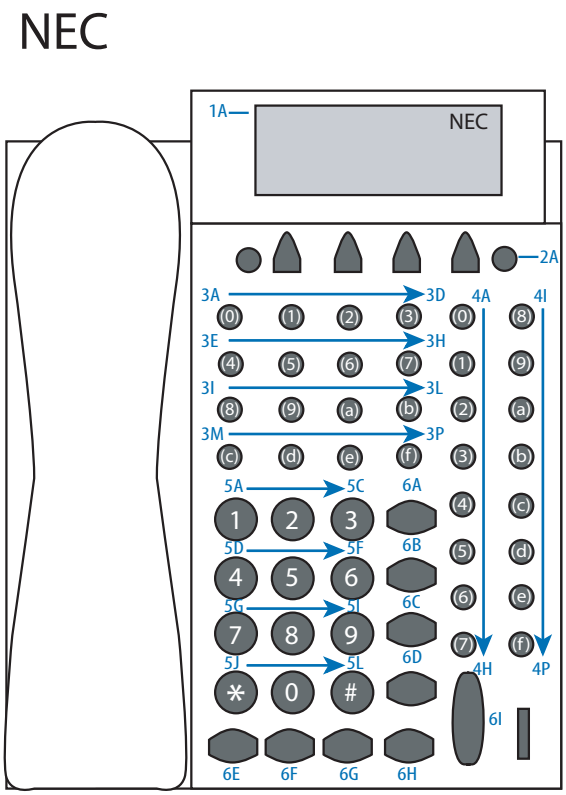
A Call from an extension connected to Ch1 to an extension connected to Ch2.

CHANNEL	EVENT	CAUSE
Channel 2	EVT_OFFHOOK	//phone (X101) picked up

```
Channel 2     EVT_MESSAGE_CHANGE (17)
Channel 2      101->[   ]
Channel 1     EVT_MESSAGE_CHANGE (17)
Channel 1
Channel 2     EVT_DIGIT_PRESSED(1)       //phone (X101) dialed "1"
Channel 2     EVT_MESSAGE_CHANGE (17)
Channel 2      101->[1  ]
Channel 2     EVT_DIGIT_PRESSED(0)       //phone (X101) dialed "0"
Channel 2     EVT_MESSAGE_CHANGE (17)
Channel 2      101->[10 ]
Channel 2     EVT_DIGIT_PRESSED(0)       //phone (X101) dialed "0"
Channel 2     EVT_MESSAGE_CHANGE (17)
Channel 2      101->[100]
Channel 1     EVT_MESSAGE_CHANGE (17)
Channel 1      100<-[101]
Channel 1     EVT_OFFHOOK                //phone (X101) picked up
Channel 1     EVT_MESSAGE_CHANGE (17)
Channel 1      100==[101]
Channel 2     EVT_MESSAGE_CHANGE (17)
Channel 2      101==[100]
Channel 2     EVT_MESSAGE_CHANGE (17)    //phone (X101) hung up
Channel 2
Channel 1     EVT_MESSAGE_CHANGE (17)
Channel 1            BUSY
Channel 2     EVT_ONHOOK
Channel 1     EVT_MESSAGE_CHANGE (17)    //phone (X100) hung up
Channel 1
Channel 1     EVT_ONHOOK
```

PHONE MAP

Nortel DTU-32D-2



1A
EVT_MESSAGE_CHANGE

2A EVT_HELP_BUTTON_PRESSED
2B EVT_HOLD_BUTTON_PRESSED

3A–3P
EVT_LINE_BUTTON_PRESSED
Subreasons: 0x00000000–0x00000010

4A–4P
EVT_FUNCTION_BUTTON_PRESSED
Subreasons: 0x00000000–0x00000010

5A–5L
EVT_DIGIT_PRESSED
Subreasons: Correspond to digit pressed

6A
EVT_FEATURE_BUTTON_PRESSED
6B
EVT_RECALL_BUTTON_PRESSED
6C
EVT_CONF_BUTTON_PRESSED
6D
EVT_REDIAL_BUTTON_PRESSED
6E
EVT_HOLD_BUTTON_PRESSED
6F
EVT_TRANSFER_BUTTON_PRESSED
6G
EVT_ANSWER_BUTTON_PRESSED
6H
EVT_SPEAKER_BUTTON_PRESSED
6I
EVT_UP_DOWN

NEC NEAX Event Sequences

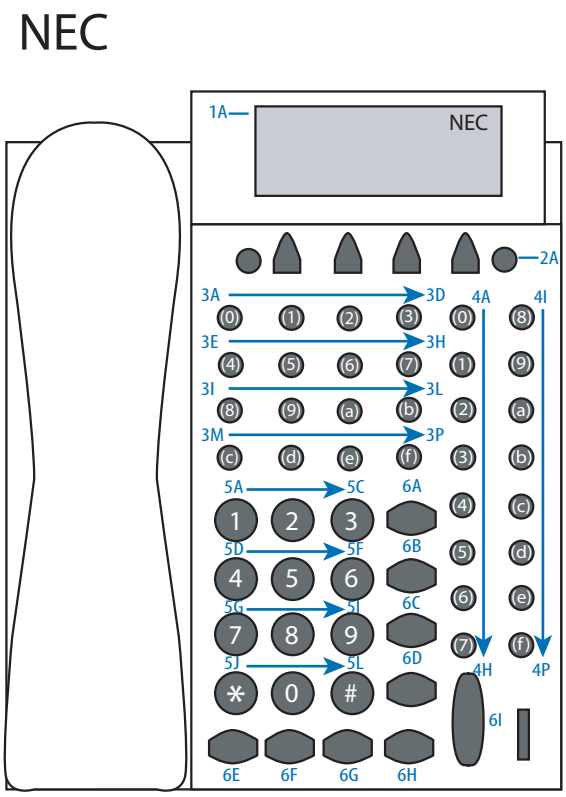
Following is a list of events that can be expected using the NGX with an NEC NEAX PBX. The recorded D-channel events have also been provided on CD to demonstrate the sequence of events.

EXAMPLE SEQUENCE

A Call from an extension connected to Channel 12 to an extension connected to Channel 13.

```
CHANNEL      EVENT                          CAUSE
Channel 12   EVT_OFFHOOK                    //phone (X133) picked up
Channel 12   EVT_AUDIO_ON(Handset 1)
Channel 12   EVT_LIGHT_ON(GREEN 15)
Channel 12   EVT_DIGIT_PRESSED(1)           //phone (X133) dialed "1"
Channel 12   EVT_DIGIT_RELEASED(1)
Channel 12   EVT_DIGIT_PRESSED(3)           //phone (X133) dialed "3"
Channel 12   EVT_DIGIT_RELEASED(3)
Channel 12   EVT_DIGIT_PRESSED(4)           //phone (X133) dialed "4"
Channel 12   EVT_DIGIT_RELEASED(4)
Channel 12   EVT_MESSAGE_CHANGE (51)
Channel 12                 1
Channel 12
Channel 12   EVT_MESSAGE_CHANGE (51)
Channel 12                 134
Channel 12
Channel 13   EVT_RING_ON(1)
Channel 12   EVT_MESSAGE_CHANGE (51)
Channel 12                 134
Channel 12    2:25 PM THU
Channel 13   EVT_LIGHT_FASTFLASHING(RED 15)
Channel 13   EVT_LIGHT_ON(RED 15)           //phone (X134) picked up
Channel 13   EVT_RING_OFF(1)
Channel 13   EVT_OFFHOOK
Channel 13   EVT_AUDIO_ON(Handset 1)
Channel 13   EVT_LIGHT_ON(GREEN 15)
Channel 13   EVT_MESSAGE_CHANGE (51)
Channel 13                 1
Channel 13
Channel 13   EVT_MESSAGE_CHANGE (51)
Channel 13                 133
Channel 13
Channel 13   EVT_MESSAGE_CHANGE (51)
Channel 13                 133
Channel 13    2:25 PM THU
Channel 12   EVT_ONHOOK                     //phone (X133) hung up
Channel 12   EVT_AUDIO_OFF(Handset 1)
Channel 12   EVT_LIGHT_OFF(GREEN 15)
Channel 13   EVT_ONHOOK                     //phone (X133) hung up
Channel 13   EVT_AUDIO_OFF(Handset 1)
Channel 13   EVT_LIGHT_OFF(GREEN 15)
Channel 13   EVT_LIGHT_OFF(RED 15)
Channel 12   EVT_MESSAGE_CHANGE (51)
Channel 12
Channel 12    2:26 PM THU   5
Channel 13   EVT_MESSAGE_CHANGE (51)
Channel 13
Channel 13    2:26 PM THU   5
```

PHONE MAP



1A
EVT_MESSAGE_CHANGE

2A EVT_HELP_BUTTON_PRESSED
2B EVT_HOLD_BUTTON_PRESSED

3A–3P
EVT_LINE_BUTTON_PRESSED
Subreasons: 0x00000000–0x00000010

4A–4P
EVT_FUNCTION_BUTTON_PRESSED
Subreasons: 0x00000000–0x00000010

5A–5L
EVT_DIGIT_PRESSED
Subreasons: Correspond to digit pressed

6A
EVT_FEATURE_BUTTON_PRESSED
6B
EVT_RECALL_BUTTON_PRESSED
6C
EVT_CONF_BUTTON_PRESSED
6D
EVT_REDIAL_BUTTON_PRESSED
6E
EVT_HOLD_BUTTON_PRESSED
6F
EVT_TRANSFER_BUTTON_PRESSED
6G
EVT_ANSWER_BUTTON_PRESSED
6H
EVT_SPEAKER_BUTTON_PRESSED
6I
EVT_UP_DOWN

Samsung Event Sequences

Following is a list of events that can be expected using the NGX with a Samsung DCS 8282 PBX.

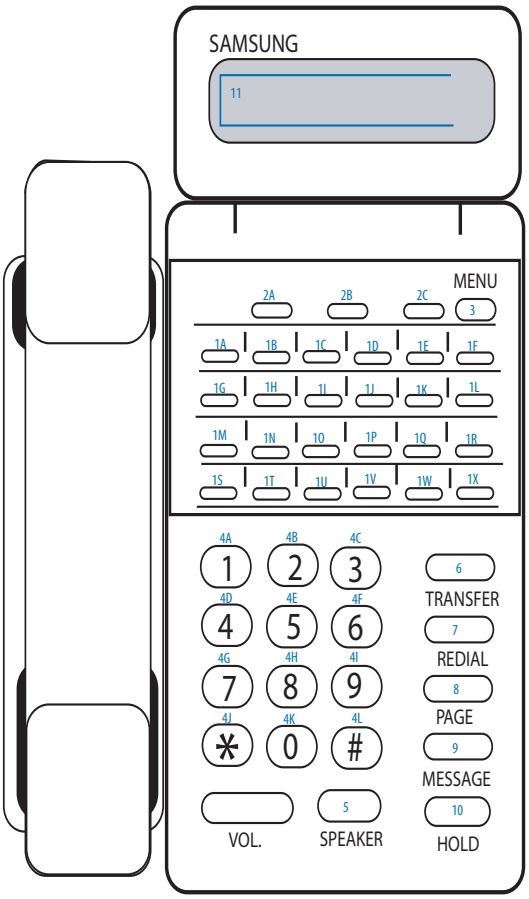
Example Sequence

A Call from an extension connected to Channel 2 to an extension connected to Channel 1.

```
Channel 2 (11efa40) EVT_OFFHOOK //phone (x201) picked up
Channel 2 (104a) EVT_AUDIO_CHANGE (0x0000000000000005)(Handset TRAN
Speaker TRAN )
Channel 2 (102a) EVT_DISPLAY_CLEAR(0x0000)
Channel 2 (1008) EVT_MESSAGE_CHANGE (52)
Channel 2 (1008)  201:
Channel 2 (1008)
Channel 2 (1008)
Channel 2 (1006) EVT_DIGIT_PRESSED(2) //phone (x201) dialed "2"
Channel 2 (102a) EVT_DISPLAY_CLEAR(0x0000)
Channel 2 (1006) EVT_DIGIT_PRESSED(0) //phone (x201) dialed "0"
Channel 2 (1006) EVT_DIGIT_PRESSED(2) //phone (x201) dialed "2"
Channel 2 (1008) EVT_MESSAGE_CHANGE (52)
Channel 2 (1008)  20
Channel 2 (1008)
Channel 2 (1008)
Channel 2 (102a) EVT_DISPLAY_CLEAR(0x0000)
Channel 1 (101f) EVT_RING_ON(0)
Channel 2 (1001) EVT_LIGHT_ON(0x0000000000000100)(GREEN 0)
Channel 2 (1033) EVT_LIGHT_VERY_FASTFLASHING(0x0217)
Channel 1 (1033) EVT_LIGHT_VERY_FASTFLASHING(0x0100)
Channel 1 (1033) EVT_LIGHT_VERY_FASTFLASHING(0x0217)
Channel 1 (102a) EVT_DISPLAY_CLEAR(0x0000)
Channel 1 (1008) EVT_MESSAGE_CHANGE (52)
Channel 1 (1008)  CALL FROM 201
Channel 1 (1008)
Channel 1 (1008)
Channel 2 (1008) EVT_MESSAGE_CHANGE (52)
Channel 2 (1008)  202:Ringing
Channel 2 (1008)   CBK  MSG
Channel 2 (1008)
Channel 1 (11efa40) EVT_OFFHOOK //phone (x202) picked up
Channel 1 (104a) EVT_AUDIO_CHANGE (0x0000000000000005)(Handset TRAN
Speaker TRAN )
Channel 2 (104a) EVT_AUDIO_CHANGE (0x0000000000000000)()
Channel 2 (102a) EVT_DISPLAY_CLEAR(0x0000)
Channel 1 (1020) EVT_RING_OFF(0)
Channel 2 (1002) EVT_LIGHT_OFF(0x0000000000000217)(RED 23)
Channel 1 (1001) EVT_LIGHT_ON(0x0000000000000100)(GREEN 0)
Channel 1 (1002) EVT_LIGHT_OFF(0x0000000000000217)(RED 23)
Channel 1 (102a) EVT_DISPLAY_CLEAR(0x0000)
Channel 2 (1008) EVT_MESSAGE_CHANGE (52)
Channel 2 (1008)  202:
Channel 2 (1008)  CONF LISTEN MUTE
Channel 2 (1008)
Channel 1 (1008) EVT_MESSAGE_CHANGE (52)
Channel 1 (1008)  201:
Channel 1 (1008)  CONF LISTEN MUTE
Channel 1 (1008)
Channel 1 (11efa40) EVT_ONHOOK //phone (x202) hung up
Channel 1 (104a) EVT_AUDIO_CHANGE (0x0000000000000000)()
```

```
Channel 2 (104a) EVT_AUDIO_CHANGE (0x0000000000000005)(Handset TRAN
Speaker TRAN )
Channel 2 (102a) EVT_DISPLAY_CLEAR(0x0000)
Channel 1 (102a) EVT_DISPLAY_CLEAR(0x0000)
Channel 1 (1008) EVT_MESSAGE_CHANGE (52)
Channel 1 (1008)  202:
Channel 1 (1008)
Channel 1 (1008)
Channel 1 (1028) EVT_DISPLAY_CLOCK(0x0000)
Channel 2 (1008) EVT_MESSAGE_CHANGE (52)
Channel 2 (1008)  202:Hang up
Channel 2 (1008)
Channel 2 (1008)
Channel 2 (11efa40) EVT_ONHOOK //phone (x202) hung up
Channel 2 (104a) EVT_AUDIO_CHANGE (0x0000000000000000)()
Channel 2 (102a) EVT_DISPLAY_CLEAR(0x0000)
Channel 2 (1008) EVT_MESSAGE_CHANGE (52)
Channel 2 (1008)  201:
Channel 2 (1008)
Channel 2 (1008)
Channel 2 (1028) EVT_DISPLAY_CLOCK(0x0000)
```

PHONE MAP



1A-1X
EVT_FUNCTION_BUTTON_PRESSED
SubReasons: 0x00000000-0x00000023

2A-2C
EVT_SOFT_BUTTON_PRESSED
SubReasons:
0x00000001 -or-
0x00000002 -or-
0x00000003

3
EVT_MENU_BUTTON_PRESSED

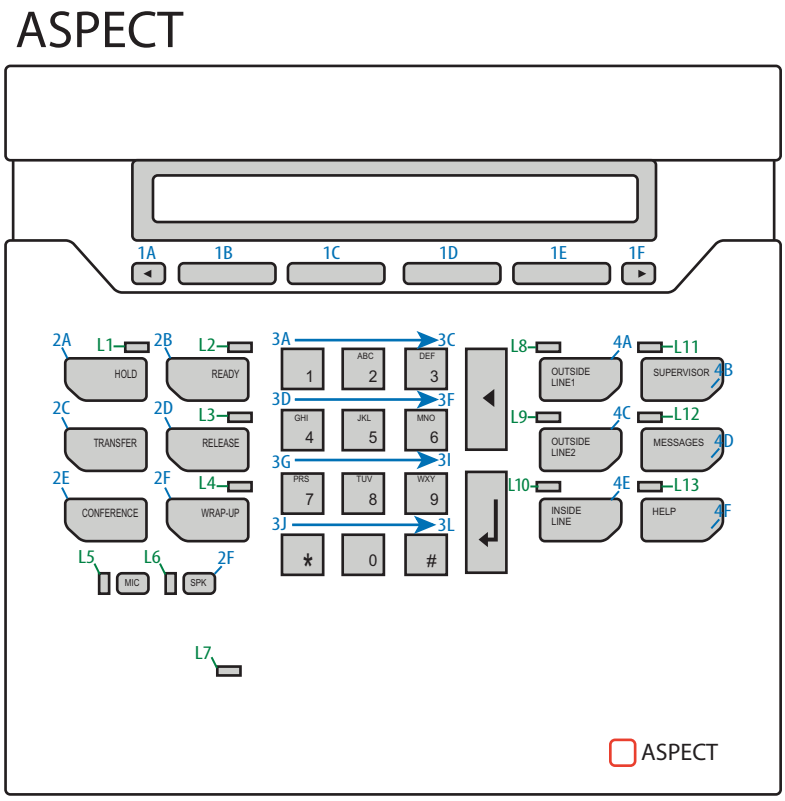
4A--4L
EVT_DIGIT_PRESSED
Subreasons: Correspond to digit pressed
* - 0x0000002A
# - 0x00000023

5
EVT_SPEAKER_BUTTON_PRESSED
Subreason: 0x00000000
6
EVT_TRANSFER_BUTTON_PRESSED
Subreason: 0x00000000
7
EVT_REDIAL_BUTTON_PRESSED
Subreason: 0x00000000
8
EVT_PAGE_BUTTON_PRESSED
Subreason: 0x00000000
9
EVT_MESSAGE_BUTTON_PRESSED
Subreason: 0x00000000
10
EVT_HOLD_BUTTON_PRESSED
Subreason: 0x00000000
11
EVT_MESSAGE_CHANGE
Subreason: 0x00000000

Aspect Event Sequences

Following is a list of events that can be expected using the NGX with a Siemens PBX. The recorded D-channel events have also been provided on CD to demonstrate the sequence of events.

Phone Map



1A–1F
EVT_SOFT_BUTTON_PRESSED
Subreasons: 0x00000000–0x00000005

2A EVT_HOLD_BUTTON_PRESSED
2B EVT_READY_BUTTON_PRESSED
2C EVT_TRANSFER_BUTTON_PRESSED
2D EVT_RELEASE_BUTTON_PRESSED
2E EVT_CONFERENCE_BUTTON_PRESSED
2F EVT_WRAPUP_BUTTON_PRESSED
2G EVT_SPEAKER_BUTTON_PRESSED

3A–3L
EVT_DIGIT_PRESSED
Subreasons: Correspond to digit pressed

4A
EVT_LINE_BUTTON_PRESSED
Subreason: 0x00000000
4B
EVT_SUPERVISOR_BUTTON_PRESSED
4C
EVT_LINE_BUTTON_PRESSED
Subreason: 0x00000001
4D
EVT_MESSAGE_BUTTON_PRESSED
4E
EVT_LINE_BUTTON_PRESSED
Subreason: 0x00000002
4F
EVT_HELP_BUTTON_PRESSED

Each light has 4 possible events: ON, OFF, FLASHING, and FASTFLASHING. The subreason listed to the right of each event remains the same for all 4 of the possible events. The subreason bit field is structured as follows: XXXX XARG where x is reserved for future use and when

A = 1 it designates an Amber light
R = 1 it designates a Red light
G = 1 it designates a Green light

L1 EVT_HOLD_LIGHT_xx Subreason : 0x00000n00
L2 EVT_READY_LIGHT_xx Subreason : 0x00000n00
L3 EVT_RELEASE_LIGHT_xx Subreason : 0x00000n00
L4 EVT_WRAPUP_LIGHT_xx Subreason : 0x00000n00
L5 EVT_MIC_LIGHT_xx Subreason : 0x00000n00
L6 EVT_SPEAKER_LIGHT_xx Subreason : 0x00000n00
L7 EVT_CALLWAITING_LIGHT_xx Subreason : 0x00000n00
L8 EVT_LINE_LIGHT_xx Subreason : 0x00000n00
L9 EVT_LINE_LIGHT_xx Subreason : 0x00000n001
L10 EVT_LINE_LIGHT_xx Subreason : 0x00000n02
L11 EVT_SUPERVISOR_LIGHT_xx Subreason : 0x00000n00
L12 EVT_MESSAGE_LIGHT_xx Subreason : 0x00000n00
L13 EVT_HELP_LIGHT_xx Subreason : 0x00000n0