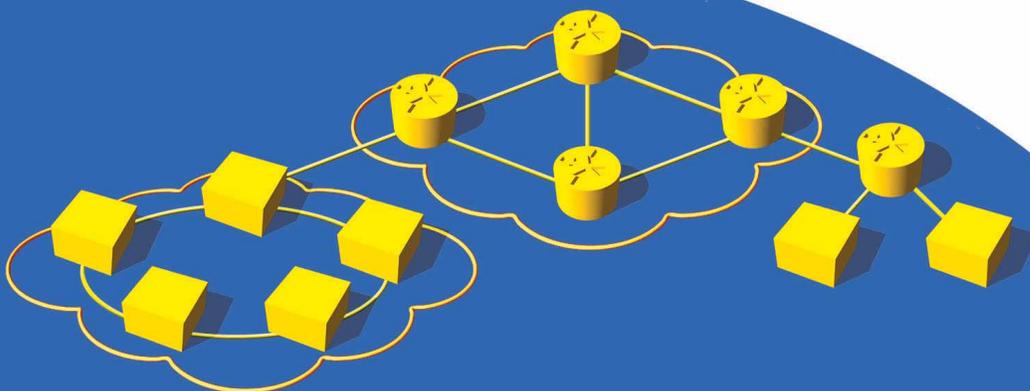


## Application Note

# Abacus Events AutoTrack

June 2004



## **Spirent Communications, Inc.**

27349 Agoura Road  
Calabasas Hills, CA  
91301 USA

### **Support Contacts**

E-mail: [support@spirentcom.com](mailto:support@spirentcom.com)  
Web: <http://support.spirentcom.com>  
Toll Free: 1-800-SPIRENT (1-800-774-7368)  
Phone: +1 818-676-2300  
Fax: +1 818-881-9154

### **Copyright**

© 2003 Spirent Communications, Inc. All Rights Reserved.

All of the company names and/or brand names and/or product names referred to in this document, in particular, the name "Spirent" and its logo device, are either registered trademarks or trademarks of Spirent plc and its subsidiaries, pending registration in accordance with relevant national laws. All other registered trademarks or trademarks are the property of their respective owners.

The information contained in this document is subject to change without notice and does not represent a commitment on the part of Spirent Communications. The information in this document is believed to be accurate and reliable, however, Spirent Communications assumes no responsibility or liability for any errors or inaccuracies that may appear in the document.

### **Warranty**

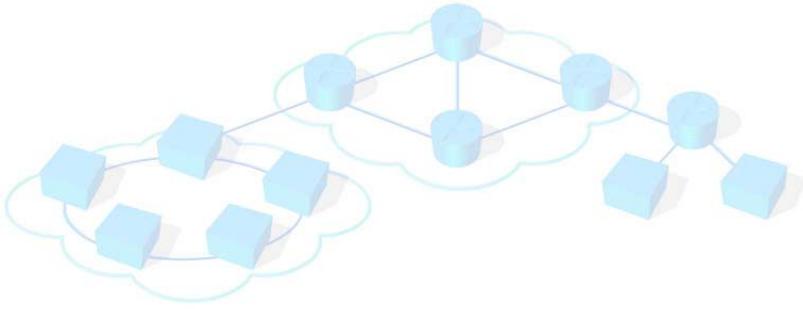
Spirent Communications, Inc. ("Spirent") warrants that its Products will conform to the description on the face of order, that it will convey good title thereto, and that the Product will be delivered free from any lawful security interest or other lien or encumbrance.

Spirent further warrants to Customer that hardware which it supplies and the tangible media on which it supplies software will be free from significant defects in materials and workmanship for a period of twelve (12) months, except as otherwise noted, from the date of delivery (the "Warranty Period"), under normal use and conditions.

To the extent the Product is or contains software ("Software"), Spirent also warrants that during the Warranty Period, the Software which it supplies will operate substantially in accordance with the specifications supplied by Spirent for such Software. Spirent does not warrant that the functions contained in the Software will meet a specific requirement or that the operation will be uninterrupted or error free. Spirent shall have no warranty obligations whatsoever with respect to any Software which has been modified in any manner by Customer or any third party. The Warranty Period for Wireless software products shall be defined as ninety (90) days from the date of delivery.

Defective Products and Software under warranty shall be, at Spirent's discretion, repaired or replaced or a credit issued to Customer's account for an amount equal to the price paid for such Product provided that: (a) such Product is returned to Spirent after first obtaining a return authorization number and shipping instructions, freight prepaid, to Spirent's location in the United States; (b) Customer provides a written explanation of the defect or Software failure claimed by Customer; and (c) the claimed defect actually exists and was not caused by neglect, accident, misuse, improper installation, improper repair, fire, flood, lightning, power surges, earthquake, or alteration. Spirent will ship repaired Products to Customer, freight prepaid, within ten (10) working days after receipt of defective Products. Except as otherwise stated, any claim on account of defective materials or for any other cause whatsoever will conclusively be deemed waived by Customer unless written notice thereof is given to Spirent within the Warranty Period. Spirent reserves the right to change the warranty and service policy set forth above at any time, after reasonable notice and without liability to Customer.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, ALL IMPLIED WARRANTIES, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, ARE HEREBY EXCLUDED, AND THE LIABILITY OF SPIRENT, IF ANY, FOR DAMAGE RELATING TO ANY ALLEGEDLY DEFECTIVE PRODUCT SHALL BE LIMITED TO THE ACTUAL PRICE PAID BY THE CUSTOMER FOR SUCH PRODUCT. THE PROVISIONS SET FORTH ABOVE STATE SPIRENT'S ENTIRE RESPONSIBILITY AND CUSTOMER'S SOLE AND EXCLUSIVE REMEDY WITH RESPECT TO ANY BREACH OF ANY WARRANTY.

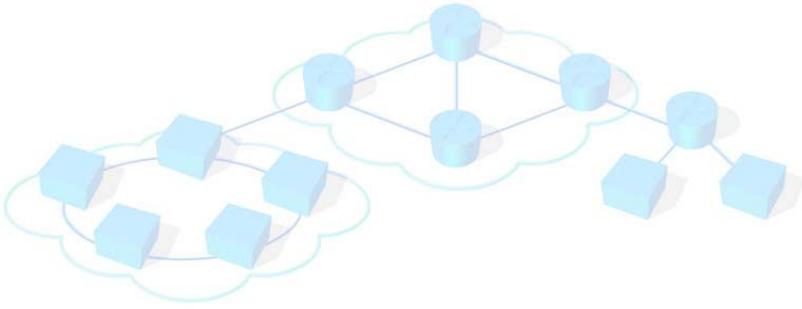


# Contents

---

- Overview ..... 2
- Introduction ..... 2
- Abacus Events ..... 3
- AutoTrack ..... 5
  - Configuration Steps ..... 5
  - Reviewing Attachments ..... 5
- Summary ..... 8





# Abacus Events AutoTrack

---

This application note provides guidelines for analyzing reported errors.

## In this document...

This application note contains the following topics:

- [Overview . . . . 2](#)
- [Introduction . . . . 2](#)
- [Abacus Events . . . . 3](#)
- [AutoTrack . . . . 5](#)
- [Summary . . . . 8](#)

## Overview

Test Engineers all over the world are working with more and more complicated test scenarios for VoIP equipment and network testing. The collection of results, pass/not pass conditions, test configurations, and lists of errors to analyze are growing exponentially with the constant addition of new technologies, interfaces, policies, and protocols. The ever-increasing capacity of equipment requires test engineers to analyze errors occurring on a large number of channels over a long period of time. The existing analyzing tools usually do not have the ability to store this huge volume of history records in memory.

Not so with Abacus. The new AutoTrack feature automatically analyzes all channels and records a history of calls. AutoTrack makes the analysis of tests, if not enjoyable, less time consuming, more productive, and more efficient.

This application note will assist you in making full use of AutoTrack. You will learn how to configure the tracking and how to view and analyze the data triggered by an event.

## Introduction

Abacus provides test and development engineers with a fully integrated IP and PSTN Telephony test system in a single platform. Abacus test methodology allows users to objectively measure Voice Quality (MOS, PSQM, PSQM+, PESQ) under real-world voice and signaling traffic load generation. Abacus provides functional testing, decodes, testing of performance, and interoperability. Abacus 5000 Media Payload formats include tones, WAV, Video (H.261 and H.263 encoding), and encoding and decoding of CODEC variants G.711, G.726, G.723.1, G.729A/B. Supported protocols include:

- PSTN protocols: SS7, CAS, MFR1/R1.5/R2, ISDN PRI, GR-303, V5.1/V5.2.
- Fax and Data Modems: T.30, T.38, V.90, V.92
- VoIP protocols: RTP/RTCP, MGCP, MEGACO, SIGTRAN, H.323 and SIP.

Multiple physical interfaces are provided to connect to the DUT/SUT:

- 10/100/1000Base-T Ethernet for IP traffic
- T1/E1/J1, G.747, and T3/E3 for digital PSTN traffic
- FXO and FXS (ground start/loop start) for analog PSTN traffic

To view and analyze the results of tests configured with all supported means, Abacus provides monitoring tools such as the **Audio Monitor**, **Data Link Monitors**, and various **Signaling Monitors**. In conjunction with these tools, **AutoTrack** delivers powerful and efficient test analysis.

To help you make full use of AutoTrack, the sections below will show you how to specify call generation errors, how to configure AutoTrack, and how to review and analyze the history provided.

## Abacus Events

Abacus provides real-time responses to the thresholds and errors specified for carried-out tests. For all errors, you should determine how Abacus should react when an error occurs. Abacus can respond by stopping the script, by stopping the test, or by freezing the errored channel.

Threshold and erroneous states you determine for your tests should be configured in the **Timeouts and Errors** GUI window. Each occurrence of a timeout or error is defined as an *Event*. Events are grouped into occurrences related to system, call generation, and switching timeouts and errors.

The error conditions are processed while the test is running for all active channels on Abacus, and errors are generated in cases of failure. You can display the errors totaled for all of the active channels or for any individual channel.

Examples of errors for call generation are shown in *Figure 1*. The example on the left shows *regular* errors, and the example on the right shows *digit* errors.

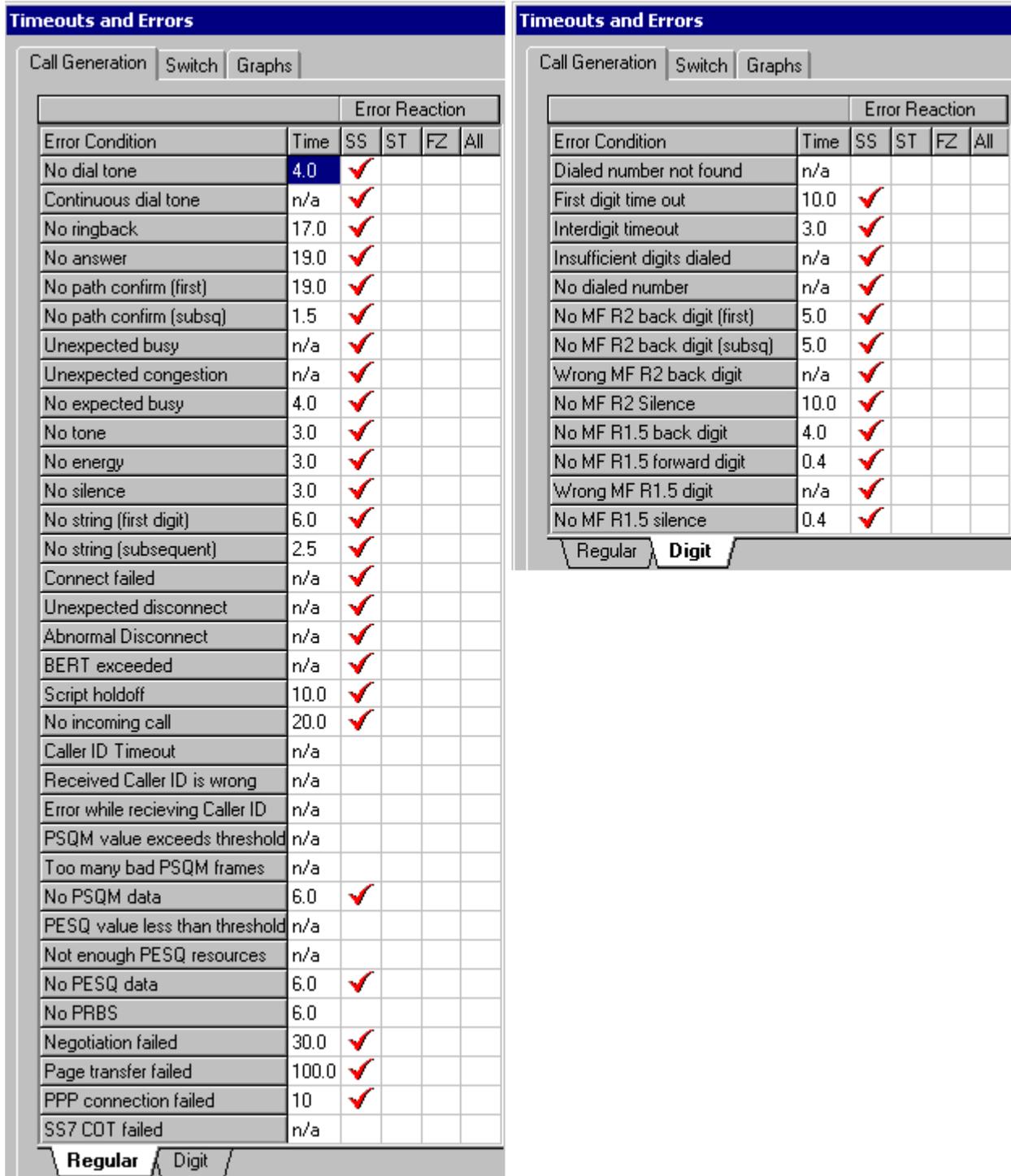


Figure 1. Examples of Errors for Call Generation

## AutoTrack

There can be tens of thousands of channels configured in a test, and the test can run for days. The major task of AutoTrack is to catch for you the signaling history only on the channels where errors occurred and to provide you with information about failures. AutoTrack provides you not only with the record of an event, but also with the history of signaling messages around the time of the event. One message before and three consecutive messages after the event are recorded. This record is called an **Attachment**. The messages in an Attachment are already decoded and put in a timely order to analyze when and why the event happened, and how the device under test (DUT) responded to the event.

To have Abacus generate Attachments, you must configure the test to do it.

## Configuration Steps

First you need to identify which set of originate/terminate channels for which you want to have attachments generated. Use the **Partition and Timing** window to select the sets, and the **Events** window to open the attachments.



### Enable and view Attachments as follows:

- 1 For all signaling types, enable the **Attachments** checkbox at the bottom of the **Partition and Timing** window.
- 2 On the **Partition and Timing | Association** window, click on a box in the **Attach** column for at least one set. A green checkmark appears.
- 3 Start the test. If an error occurs on one or more channels in the selected set(s), an Attachment will accompany it.
- 4 Open the **Events** window. If an error has occurred, an envelope icon is displayed in the **Attach** column.
- 5 Click on the envelope icon to view the Attachment.

*Figure 2* shows the **Partition and Timing** window with **Attachments** enabled.

## Reviewing Attachments

### Events Window

A history of all events that happened during the test is displayed in the **Events** window. The **Events** window provides the following information:

- Real time of event occurrence
- Channel type and channel number
- Action number that identifies the position of the action relative to the start of your script.

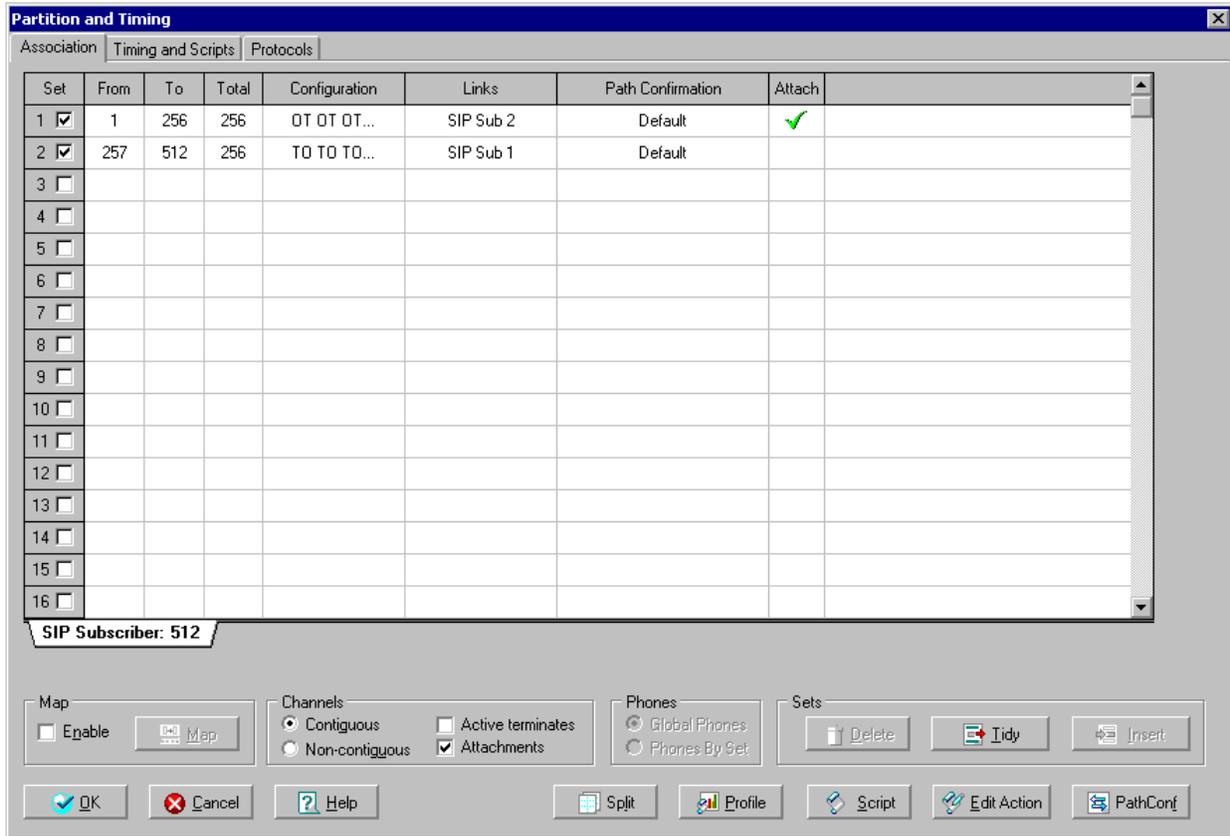


Figure 2. Partition and Timing Window

- A description of the event
- Comments describing functionality of certain actions
- Phone number associated with the channel
- Reasons for the error
- Attachment file with AutoTrack data

The events are shown in chronological order, with the most recent event appearing at the bottom of the list. Newly received events are added to the list as they occur.

If the test has generated a large number of events, it can take several minutes before the window is displayed. The events of a test are saved in a log file that can be viewed later.

If **Attachment** is configured, an envelope icon appears in the **Attach** column in the **Events** window when the event occurs.

An example of **Events** window for SIP is shown in [Figure 3](#), and an example for T1 is shown in [Figure 4](#).

| Elapsed    | Channel     | Act | Event                   | Comment                                   | Phone                       | Cause | Attach |
|------------|-------------|-----|-------------------------|---|-----------------------------|-------|--------|
| *000:00:00 |             |     | Test Started            |   |                             |       |        |
| 000:00:09  | SIP Sub 2   | Trm | No path confirm (first) |   | 4002:14085411002@10.2.16.11 |       | ✉      |
| 000:00:10  | SIP Sub 4   | Trm | No path confirm (first) |   | 4004:14085411004@10.2.16.11 |       | ✉      |
| 000:00:11  | SIP Sub 6   | Trm | No path confirm (first) |   | 4006:14085411006@10.2.16.11 |       | ✉      |
| 000:00:12  | SIP Sub 258 | 1   | Unexpected disconnect   | A calls B, VolP, confirms for Call Length | 4258:14085411258@10.2.16.12 |       |        |
| 000:00:12  | SIP Sub 8   | Trm | No path confirm (first) |   | 4008:14085411008@10.2.16.11 |       | ✉      |
| 000:00:13  | SIP Sub 260 | 1   | Unexpected disconnect   | A calls B, VolP, confirms for Call Length | 4260:14085411260@10.2.16.12 |       |        |
| 000:00:13  | SIP Sub 10  | Trm | No path confirm (first) |   | 4010:14085411010@10.2.16.11 |       | ✉      |
| 000:00:14  | SIP Sub 262 | 1   | Unexpected disconnect   | A calls B, VolP, confirms for Call Length | 4262:14085411262@10.2.16.12 |       |        |
| 000:00:14  | SIP Sub 12  | Trm | No path confirm (first) |   | 4012:14085411012@10.2.16.11 |       | ✉      |
| 000:00:15  | SIP Sub 264 | 1   | Unexpected disconnect   | A calls B, VolP, confirms for Call Length | 4264:14085411264@10.2.16.12 |       |        |
| 000:00:15  | SIP Sub 14  | Trm | No path confirm (first) |   | 4014:14085411014@10.2.16.11 |       | ✉      |
| 000:00:16  | SIP Sub 266 | 1   | Unexpected disconnect   | A calls B, VolP, confirms for Call Length | 4266:14085411266@10.2.16.12 |       |        |
| 000:00:16  | SIP Sub 16  | Trm | No path confirm (first) |   | 4016:14085411016@10.2.16.11 |       | ✉      |
| 000:00:17  | SIP Sub 268 | 1   | Unexpected disconnect   | A calls B, VolP, confirms for Call Length | 4268:14085411268@10.2.16.12 |       |        |
| 000:00:17  | SIP Sub 18  | Trm | No path confirm (first) |   | 4018:14085411018@10.2.16.11 |       | ✉      |
| 000:00:18  | SIP Sub 270 | 1   | Unexpected disconnect   | A calls B, VolP, confirms for Call Length | 4270:14085411270@10.2.16.12 |       |        |
| 000:00:18  | SIP Sub 20  | Trm | No path confirm (first) |   | 4020:14085411020@10.2.16.11 |       | ✉      |
| 000:00:19  | SIP Sub 272 | 1   | Unexpected disconnect   | A calls B, VolP, confirms for Call Length | 4272:14085411272@10.2.16.12 |       |        |
| 000:00:19  | SIP Sub 22  | Trm | No path confirm (first) |   | 4022:14085411022@10.2.16.11 |       | ✉      |
| 000:00:20  | SIP Sub 274 | 1   | Unexpected disconnect   | A calls B, VolP, confirms for Call Length | 4274:14085411274@10.2.16.12 |       |        |

Figure 3. Events Window for SIP

| Elapsed    | Channel  | Act | Event                   | Comment             | Phone | Cause | Attach |
|------------|----------|-----|-------------------------|---------------------|-------|-------|--------|
| *000:00:00 |          |     | Test Started            |                     |       |       |        |
| 000:00:24  | T1 Sub 1 | 1   | No path confirm (first) | A calls B, C7771001 |       |       | ✉      |
| 000:00:25  | T1 Sub 2 | 1   | No path confirm (first) | A calls B, C7771002 |       |       | ✉      |
| 000:00:26  | T1 Sub 3 | 1   | No path confirm (first) | A calls B, C7771003 |       |       | ✉      |
| 000:00:27  | T1 Sub 4 | 1   | No path confirm (first) | A calls B, C7771004 |       |       | ✉      |
| 000:00:28  | T1 Sub 5 | 1   | No path confirm (first) | A calls B, C7771005 |       |       | ✉      |

Figure 4. Events Window for T1

## Attachment Review

To view the history of signaling flow around the event, double-click on an envelope icon. The history screen with decoded messages appears.

*Figure 5* shows the **Events** window with an Attachment open for SIP, and *Figure 6* shows the **Events** window with an Attachment open for SS7. The failed call is shown in red.

You can see from these examples how the messages flow relates to events. In the SS7 example, you see the message “Release” because of “Unexpected Disconnect” error. This error could happen, for example, if the device under test (DUT) lost connection to Abacus or dropped the call.

In the SIP example, the ACK message got through, but the path for voice was lost at 1min 48sec (elapsed time). An error “No Path Confirmation” was generated. The originate channel disconnects the call and sends a BYE message, and then receives OK from the terminate side. It indicates that the signaling worked correctly, but RTP processing failed on the DUT.

If during the three calls after the reviewed event occurs, new events come to the **Events** window on the same channel, they will appear in the same Attachment. No new Attachments will be generated for such events, to avoid an infinite list of Attachments. After the duration of the three calls after the errored event has expired, AutoTrack will process the next events, and a new Attachment can be opened.

## Summary

This application note has shown how you can utilize additional Abacus analyzing capabilities to make your testing or development more productive. You learned how to configure your test to use the most important functions of AutoTrack:

- Help you to concentrate on analyzing only the small number of failed channels, instead of going through thousand and thousands recorded calls.
- Present a history of messages surrounding the event.

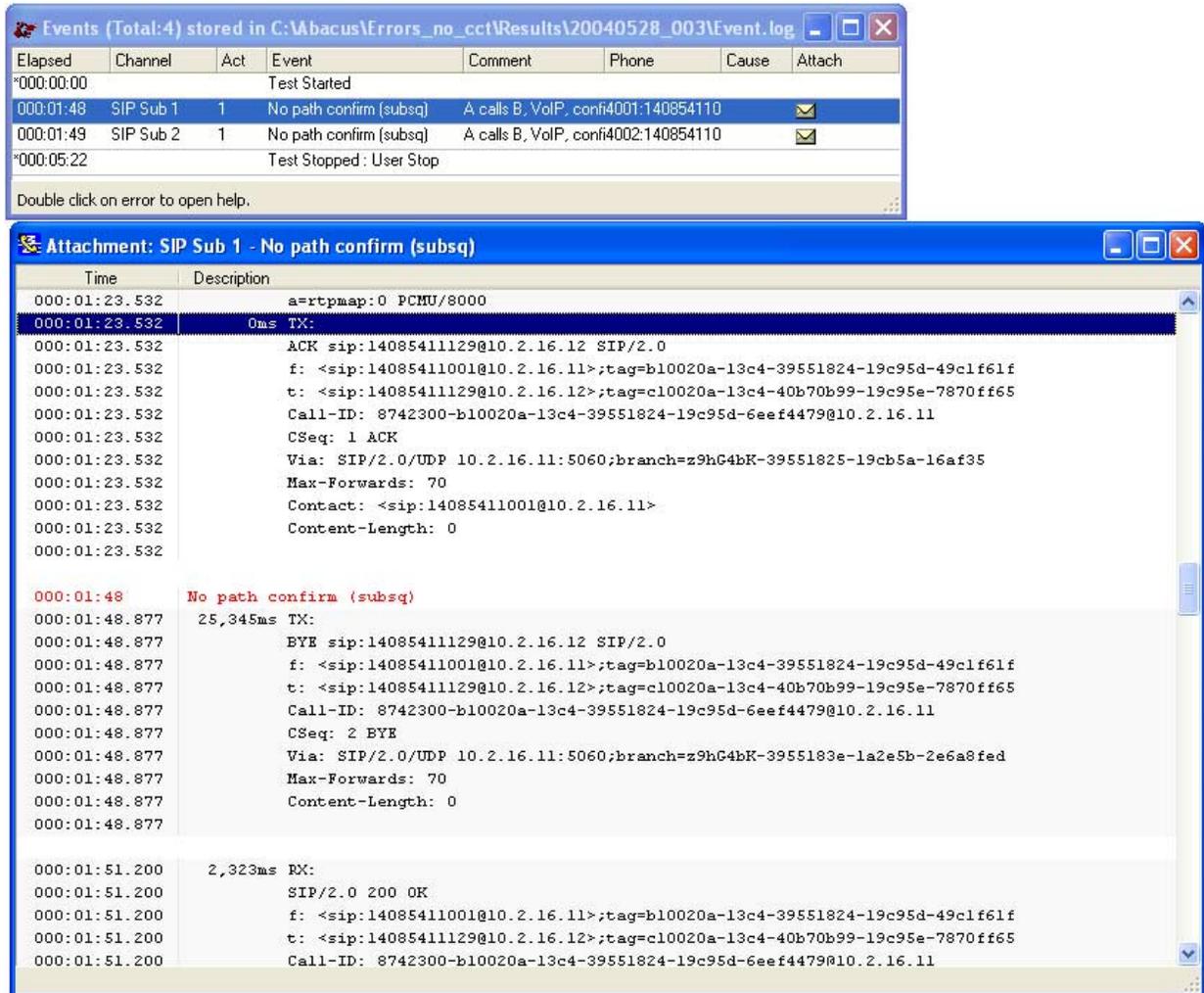


Figure 5. Events Window with Attachment for SIP

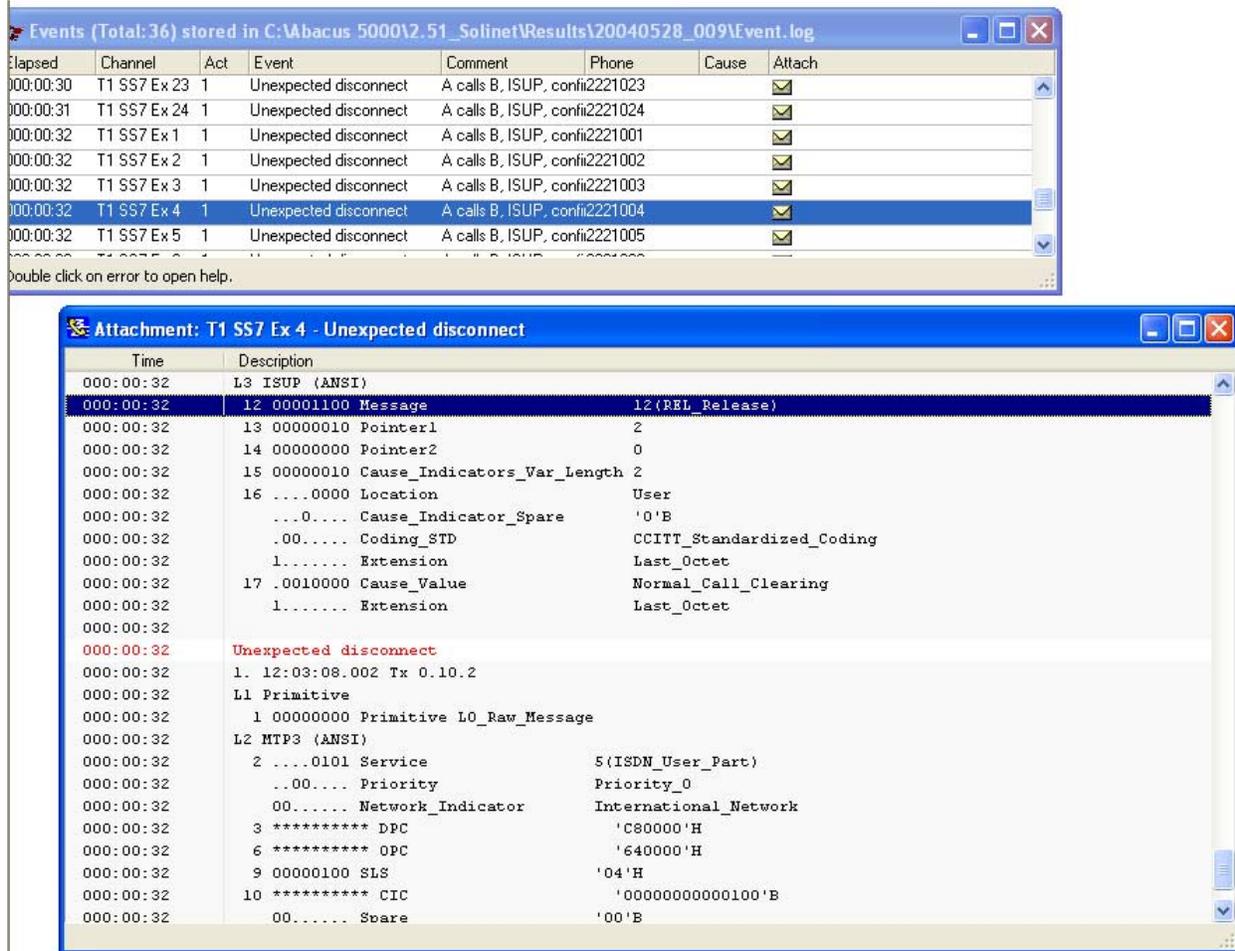


Figure 6. Events Window with Attachment for SS7