Dialogic® Brooktrout® TR1000 Series SDK

Installation and Configuration Guide

Release 5.2.0

Document Number 931-131-03

Copyright and Legal Disclaimer

International Notice

Due to differing national regulations and approval requirements, certain Dialogic® products are designed for use only in specific countries, and may not function properly in a country other than the country of designated use. As a user of these products, you are responsible for ensuring that the products are used only in the countries for which they were intended. For information on specific products, contact Dialogic Corporation on the web at ***www.dialogic.com*** or:

9800 Cavendish Blvd.
5th Floor
Montreal, Quebec, Canada
H4M 2V9
7814494100

Technical Support

For Technical Support, see ***Getting Technical Support on page xxxvi*** and the web:

www.dialogic.com/support

Limited Warranty

Dialogic Corporation (“Dialogic”) warrants the hardware component of the product described in this documentation (the “Product”) to be free from defects in materials and workmanship under normal and proper use for the period specified in your signed agreement with Dialogic. In the event that you have no signed agreement setting out a warranty period, the period shall be the standard warranty period for such hardware component of the product set out on www.dialogic.com on the date of your purchase of the product. Dialogic also warrants the disk on which software and firmware are recorded to be free from defects in materials and workmanship under normal and proper use for a period of 90 days from the date of purchase from Dialogic. ***This warranty does not apply to the software and firmware themselves.*** This warranty also does not apply to any expendable components, any damage resulting from abuse of the Product, or normal wear and tear. In the event of a warranty claim, the item, if in the opinion of Dialogic it is proved to be defective, will be repaired or replaced with a functionally equivalent item, at Dialogic's sole option, upon delivery to Dialogic of the defective item, together with a dated proof of purchase and specification of the problem. Dialogic is not responsible for transportation and related charges in connection with shipment of items to Dialogic for warranty service. Dialogic reserves the right to charge for inspection at Dialogic's then prevailing rates of returned items if it is determined that the items were not defective within the terms of the warranty. To obtain warranty service return the Product, contact Dialogic Technical Support.

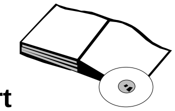
With respect to software and firmware, it should be understood that these components are complex works which may contain undiscovered defects. Although the software and firmware provided with the Product contain substantially the features described in the documentation, to the extent applicable to the product purchased, Dialogic does not warrant that the operation of such software and firmware will meet the user’s requirements or be uninterrupted or free of errors.

No oral or written information or advice given by Dialogic or its authorized representatives will create a warranty or increase the scope of this warranty. No representative, agent, dealer or employee of Dialogic is authorized to give any other warranty or to assume for Dialogic any other liability in connection with the sale and service of the Product. ***Except as expressly agreed by Dialogic in writing, Dialogic makes no representations or warranties of any kind, express or implied, with respect to the Product or any hardware, software or firmware components thereof. In particular, but without limitation of the foregoing, Dialogic disclaims all implied warranties of merchantability or fitness for a particular purpose and there are no warranties that extend beyond the description or duration of this warranty.*** Some states or countries do not allow the exclusion of implied warranties so the above exclusion may not apply to you.

In no event shall Dialogic be liable for loss of profits or indirect, special, incidental, or consequential damages arising out of the use of or inability to use the Product. The sole and exclusive remedy, in contract, tort or otherwise, available for a breach of this warranty and for any and all claims arising out of or in any way connected with the purchase of the Product shall be limited to the repair or replacement of any defective item or, at Dialogic's sole option, the payment of actual direct damages not to exceed the payments made to Dialogic for the Product in question. Some states do not allow the limitation or exclusion of liability for incidental or consequential damages, so the above limitation and exclusion may not apply to you.

This warranty gives you specific legal rights. You may also have other rights which vary from state-to-state or country-to-country. Any provision of this warranty that is prohibited or unenforceable in any jurisdiction shall, as to such jurisdiction, be ineffective to the extent of such prohibition or unenforceability without invalidating the remaining provisions hereof or affecting the validity of enforceability of such provision in any other jurisdiction.

Bfv API Documentation Roadmap



Start

TR1000 Series SDK Installation and Configuration Guide

Quick Start
Installing the Developer Kit on Redhat Linux
Installing the Developer Kit on Solaris
Installing the Developer Kit on Windows
Installing the Device Driver on Windows
Configuring Software
Directory Structure
AccuCall and the AccuCall Wizard
Using the TECUpdate Utility
Using the Brooktrout License Manager to Activate Brooktrout Products
Downloading Firmware
Compliance Information
Configuration from a Custom Installation Program

TR1000 Series SDK Developer Guide

Developing Applications Using the Bfv API
Debugging
Sample Applications and Utilities
Transferring Calls
Automatic Speech Recognition (ASR)
Managing Fax Over IP Sessions
Robbed Bit Signaling
ISDN Call Processing & Management
Using R2 Signaling Capability of BSMI
Packaging Your Application for Windows
G3 Legacy Utilities
Recompiling on Linux Platforms
Compiling On Linux Platforms

Bfv API Reference Manual Volumes 1 - 6

Bfv API Reference Manual
Volume 1
Administration, Management, and Configuration

- Bfv API Overview
- Administration and Initialization
- Firmware
- Configuration, Status and Monitoring
- Debugging, Error Handling and Return Values
- Miscellaneous Functions

Bfv API Reference Manual
Volume 2
Bfv-Level Call Control and Call Switching

- Bfv API Overview
- Call Control Overview
- Bfv-Level Call Control
- Dialing Database Functions
- Data Structures

Bfv API Reference Manual
Volume 3
Media Processing

- Bfv API Overview
- Signal Generation and Detection
- Voice Play/Record
- Infopkt File Functions
- Audio Conferencing
- Audio Conferencing Functions
- Audio Conferencing Programming Examples

Bfv API Reference Manual
Volume 4
Fax Processing

- Bfv API Overview
- Fax Overview
- Fax Functions
- TIFF-F Files Functions

Bfv API Reference Manual
Volume 5
BSMI-Level Call Control and Call Switching

- Bfv API Overview
- BOSTON Simple Message Interface (BSMI)
- BSMI General Message Structure
- R2 Signaling Protocol with BSMI
- LEC Protocols with BSMI
- Host to Module (L4L3m) Messages
- Module to Host (L3L4m) Messages
- B-Channel and D-Channel Maintenance

Bfv API Reference Manual
Volume 6
Appendices

- **App A** - Configuration Files
- **App B** - Bfv API Structures
- **App C** - Hangup Codes
- **App D** - BSMI and ISDN Cause Codes
- **App E** - Infopkt Parameter Values
- **App F** - Call Progress Notes
- **App G** - Country-Specific Parameter Files
- **App H** - Deprecated and Unsupported Functionality

SOFTWARE DEVELOPER LICENSE AGREEMENT

*** IMPORTANT ***

PLEASE READ CAREFULLY BEFORE OPENING THE ENCLOSED PACKAGE.

This Software Developer License Agreement is a legal agreement between Dialogic Corporation, and its affiliates or subsidiaries (collectively, "Dialogic") and you (either an individual or an entity) ("Developer"). ***IF YOU HAVE A WRITTEN, SIGNED LICENSE AGREEMENT GOVERNING THE USE OF THE ACCOMPANYING SOFTWARE, THE TERMS AND CONDITIONS OF THAT AGREEMENT WILL APPLY TO YOUR USE OF THE SOFTWARE***. ***IF YOU HAVE WRITTEN, SIGNED SPECIAL TERMS GOVERNING THE USE OF THE ACCOMPANYING SOFTWARE ("SPECIAL TERMS"), THOSE SPECIAL TERMS ARE HEREBY INCORPORATED INTO THIS AGREEMENT.*** Read the following terms and conditions carefully before using the accompanying software. They define your rights and obligations with respect to the enclosed Software. If you do not agree to the terms of this Software Developer License Agreement, promptly return the Developer License Key that you obtained for the Software or the Dialogic® Hardware that you purchased for use with the Software, if any, to the place you obtained it for a full refund. USING THE SOFTWARE INDICATES YOUR AGREEMENT WITH AND ACCEPTANCE OF THE FOLLOWING TERMS AND CONDITIONS. AN INDIVIDUAL WHO DOES NOT HAVE AUTHORITY TO BIND THE ENTITY USING THE SOFTWARE SHOULD NOT USE THE SOFTWARE WITHOUT OBTAINING APPROVAL OF THIS AGREEMENT FROM A PERSON HAVING SUCH AUTHORITY.

YOU UNDERSTAND THAT IF YOU PURCHASED THE PACKAGE FROM AN AUTHORIZED RESELLER OF DIALOGIC, THAT RESELLER IS NOT DIALOGIC'S AGENT AND IS NOT AUTHORIZED TO MAKE ANY REPRESENTATIONS, CONDITIONS OR WARRANTIES, STATUTORY OR OTHERWISE, ON DIALOGIC'S BEHALF NOR TO VARY ANY OF THE TERMS OR CONDITIONS OF THIS AGREEMENT. IN ADDITION, YOU ACKNOWLEDGE THAT, UNLESS OTHERWISE AGREED BY THAT RESELLER IN WRITING OR PROHIBITED BY LAW, THE LIMITATIONS OF CONDITIONS AND WARRANTIES, STATUTORY OR OTHERWISE, AND LIABILITY SET FORTH IN THIS AGREEMENT ALSO APPLY TO AND BENEFIT THAT RESELLER.

The Software may include object code versions of third party software packages, including one or more of the following, which are subject to the terms and conditions of the applicable license agreements found at the websites identified below. Please refer to the attached Exhibit A to find out which, if any, licenses are applicable.

- ***ACE™ software and TAO™ software are copyrighted by Douglas C. Schmidt and his research group at Washington University, University of California, Irvine, and Vanderbilt University Copyright (c) 1993-2003, all rights reserved. More information is available at:***

    http://www.cs.wustl.edu/~schmidt/ACE.html

- ***wxWindows software is copyrighted by Julian Smart, Robert Roebling et al. Copyright (c) 1998 Julian Smart, Robert Roebling et al. More information is available at:***

    http://www.wxWindows.org

1. Definitions

1.1. "Dialogic Hardware" means voice and/or fax processing hardware and/or any other computer hardware manufactured by or for Dialogic which may (without limitation) be marketed under the brands Dialogic, Brooktrout, Excel, and/or Snowshore and purchased by the End User from Dialogic or one of its authorized resellers.

1.2. "Developer License Key" means an access code or activation key provided by Dialogic to enable a Developer to use the Software for the purposes set forth herein.

1.3. "<u>End User</u>" means the ultimate user of the Software who has obtained such Software pursuant to an End User Agreement.

1.4. "<u>End User Agreement</u>" means a sublicense agreement granting an End User the right to use the Software internally but not to further distribute or sublicense such Software, which agreement meets the requirements of Section 2.8.

1.5. "<u>Integrated Product</u>" means the Software as integrated with one or more of Developer's products to provide enhanced functionality.

1.6. "<u>License Key</u>" means an access code or activation key provided by Dialogic to enable an End User to use the Software.

1.7. "<u>Software</u>" means the software provided to you in the enclosed package, or enabled by the enclosed Developer License Key together with any accompanying documentation.

1.8. "<u>Sublicensee</u>" means any reseller, dealer, distributor or other entity authorized by Developer pursuant to a Sublicense Agreement meeting the requirements of Section 2.9, to sublicense the use of and distribution of the Software to End Users or other intermediate parties, such as dealers, in the distribution chain to End Users.

2. License

2.1. <u>Demonstration or Evaluation</u>. If the Software is provided to Developer as a demonstration or evaluation unit, (a) the licenses granted in Section 2.2 shall not apply and (b) Dialogic grants the Developer, and the Developer accepts, a non-exclusive, non-transferable right and license (without the right to grant sublicenses) to use the Software for a period of ninety (90) days from the date of delivery of the Software to Developer for the sole purpose of evaluating the suitability of such Software. Under no circumstances shall the Software be used for any commercial purposes, and Developer is not being granted any rights to distribute the Software under this Agreement without the express written permission of Dialogic.

2.2. <u>License Grant</u>. Subject to the terms and conditions contained herein, Dialogic grants Developer a license to use and distribute the Software subject to and in accordance with the following provisions.

a. If the Software is distributed to Developer for use with Dialogic Hardware, Dialogic grants Developer a royalty-free, non-transferable license (without the right to grant sublicenses, except as set forth in Section 2.9) to (i) use the Software solely for the purposes of developing Integrated Products; (ii) make copies of the Software for distribution to End Users, whether as a stand-alone product or as part of an Integrated Product, solely for use by such End Users for their internal business purposes in connection with Dialogic Hardware; and (iii) distribute the Software to End Users, whether as a stand-alone product or as part of an Integrated Product, solely for use by such End Users for their internal business purposes in connection with Dialogic Hardware. The Software is used in connection with Dialogic Hardware if it is operated only on the central processing unit of a computer served by one or more items of Dialogic Hardware and, where appropriate in connection with such use, it is downloaded into memory located on Dialogic Hardware.

b. If the Software is distributed to Developer as host-based media processing software, Dialogic grants Developer a royalty-bearing, non-transferable license (without the right to grant sublicenses, except as set forth in Section 2.9) to (i) use the Software solely for the purpose of developing Integrated Products; (ii) make copies of the Software for distribution to End Users, whether as a stand-alone product or as part of an Integrated Product for the End Users' internal business purposes; and (iii) distribute copies of the Software to End Users, whether as a stand-alone product or as part of an Integrated Product for the End Users' internal business purposes, subject to the purchase by the End User of a License Key. Developer shall not distribute or otherwise transfer any Developer License Key for any purpose without Dialogic's prior written consent.

2.3. Copies. Developer shall not remove or alter any copyright or other proprietary notices that appear on the Software and shall include all such notices on all copies of the Software (including without limitation all copies of the Software that are incorporated into Integrated Products).

2.4. No Reverse Engineering. Developer shall have no rights to any source code for the Software. Developer agrees that it shall not cause or permit the disassembly, decompilation or reverse engineering of the Software or otherwise attempt to gain access to the source code to the Software. If applicable law requires access to source code for some purpose, such as inter-operability with other software, and Developer desires access for that required purpose, Developer shall notify Dialogic, and Dialogic shall have the option, in its discretion, to (i) perform the work to derive any required information at Dialogic's usual consulting rate, or (ii) allow Developer access to source code solely for the legally required purpose.

2.5. Additional Restrictions. Developer shall not, and shall not permit any third party to disclose the results of any benchmark test of the Software to a third party without the prior written approval of Dialogic.

2.6. Reservation of Rights. Dialogic and its licensors reserve all rights not otherwise expressly granted in this Section 2. Any use, modification, reproduction or distribution of the Software not expressly permitted under this Agreement is hereby expressly prohibited.

2.7. License Keys. The use of the Software may require a License Key. Developer shall not (a) use any access code or activation key other than the Developer License Key (if any) provided by Dialogic or (b) develop or alter any access code for the Software or otherwise circumvent or attempt to circumvent the License Key or Developer License Key.

2.8. End User Agreements. The Software shall be distributed to each End User under an End User Agreement entered into by such End User which may be a shrink-wrap, break-the-seal or click-wrap or executed written agreement which provides for the same level of protection of Dialogic's title, rights and defenses as this Agreement, and pursuant to which the End User agrees (a) to use the Software solely for its own internal business purposes, and not for further distribution or licensing; (b) not to copy the Software in whole or in part, except as permitted under applicable law, and then only with the inclusion of all copyright, proprietary and other notices; (c) not to permit any parent, subsidiaries, affiliated entities or third parties to use the Software; and (d) not rent, lease, loan or transfer the Software. Where the Software or Integrated Product is distributed to an End User pursuant to Section 2.2(a), the End User shall also agree to use the Software solely in connection with Dialogic Hardware. Developer may include in such End User Agreement, the terms of the limited warranty attached as Exhibit B hereto.

2.9. Sublicenses. Developer shall have the right to sublicense its rights under Sections 2.2(a)(iii) and 2.2(b)(iii) solely to allow Sublicensees to distribute the Software to End Users, whether as a stand-alone product or as part of Developer's Integrated Product, but in no event shall any Sublicensee have the right to (a) use the Software to develop its own integrated products or (b) make copies of the Software. Each Sublicensee shall execute a Sublicense Agreement that contains substantially similar terms and conditions as are contained in this Agreement and provides for the same level of protection of Dialogic's title, rights and defenses as this Agreement.

2.10. Protection of Rights. Developer will use the same efforts that it would use to protect its own rights, including its intellectual property rights, but not less than commercially reasonable efforts to ensure that Sublicensees and End Users comply with the terms of their respective agreements, and will notify Dialogic of any known violation. Upon Dialogic's request, Developer will supply Dialogic with a copy of any sublicense agreement by and between Developer and any Sublicensees or End Users.

3. ***Ownership***

3.1. Dialogic or its licensors own and shall retain all proprietary rights, including all patent, copyright, trade secret, trademark and other intellectual property rights, in and to the Software and any modifications thereto made by any entity.

3.2. Except as set forth in a separate agreement between Dialogic and Developer, Developer shall have no right to receive any corrections, enhancements, upgrades or other modifications to the Software ("Changes"). Unless otherwise agreed to in such separate agreement, any Changes to the Software provided by Dialogic to Developer in accordance with the separate agreement shall be governed by the terms of this Agreement. If Developer does receive Changes from Dialogic, it shall incorporate those Changes into Software that it distributes to End Users under Section 2.2 for initial installation by End Users. Developer acknowledges that the license granted under this Agreement does not provide Developer with title to or ownership of the Software, but only a right of limited use under the terms and conditions of this Agreement. Developer shall keep the Software free and clear of all claims, liens and encumbrances.

4. ***Limited Warranty***

4.1. <u>Scope of Limited Warranty</u>

a. Dialogic warrants to Developer that for a period of ninety (90) days from delivery of the Software to Developer (the "Warranty Period"), the media on which the Software is delivered shall be free from defects in materials and workmanship. Dialogic's sole liability and Developer's exclusive remedy under this limited warranty shall be the replacement of the non-conforming media.

b. The remedies set forth in Section 4.1(a) are available only if Dialogic is promptly notified in writing, within the Warranty period, upon discovery of the defect by Developer.

<u>4.2.</u> <u>Disclaimer of Any Other Warranties</u>

a. NOTWITHSTANDING ANYTHING IN THIS AGREEMENT TO THE CONTRARY, ANY SOFTWARE PROVIDED UNDER SECTION 2.1 AS A DEMONSTRATION OR EVALUATION UNIT IS BEING PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND AND LICENSOR HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS OR IMPLIED, ORAL OR WRITTEN, WITH RESPECT TO THE LICENSED SOFTWARE INCLUDING, WITHOUT LIMITATION, ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE AND NON-INFRINGEMENT.

b. EXCEPT FOR THE EXPRESS, LIMITED WARRANTY PROVIDED IN SECTION 4.1, DIALOGIC MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THE SOFTWARE, AND DIALOGIC DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. DIALOGIC DOES NOT WARRANT THAT THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT IT WILL SATISFY DEVELOPER'S REQUIREMENTS.

4.3. Dialogic will not be responsible to any End User for any warranties made by Developer with respect to any Software beyond the limited warranty described in Exhibit B.

5. ***Limitation of Liability***

5.1. IN NO EVENT SHALL DIALOGIC'S OR ITS LICENSORS' AGGREGATE LIABILITY UNDER THIS AGREEMENT EXCEED THE AMOUNTS PAID BY DEVELOPER TO DIALOGIC FOR THE SOFTWARE (as "Software" is defined above).

5.2. IN NO EVENT WILL DIALOGIC OR ITS LICENSORS BE LIABLE TO DEVELOPER OR ANY END USER OR OTHER THIRD PARTY FOR SPECIAL, INCIDENTAL, CONSEQUENTIAL, EXEMPLARY, PUNITIVE, MULTIPLE OR OTHER INDIRECT DAMAGES, OR FOR LOSS OF PROFITS, LOSS OF DATA OR LOSS OF USE DAMAGES, ARISING OUT OF THIS

AGREEMENT, WHETHER BASED UPON WARRANTY, CONTRACT, TORT, STRICT LIABILITY OR OTHERWISE, EVEN IF DIALOGIC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR LOSSES.

6. Term and Termination

6.1. Termination. Dialogic shall have the right to terminate this Agreement and the license granted herein in the event Developer fails to comply with any of the terms and conditions of this Agreement and such default has not been cured within thirty (30) days after written notice of such default.

6.2. Effect of Termination. Upon termination of this Agreement, all licenses and other rights granted to you under this Agreement will become null and void, except that the End User Licenses and any Sublicenses shall survive termination as long as the End Users and Sublicensees are in compliance with the terms of this Agreement. Within five (5) days after termination of this Agreement, you shall return to Dialogic or destroy, at Developer's expense, the Software, including all copies thereof. Sections 2.3-2.8, 2.10, 3, 4, 5, 6, 7, 8, and 9 of this Agreement shall survive termination of this Agreement.

7. U.S. Government Restricted Rights

If Software is acquired by or on behalf of a unit or agency of the United States government, the Software is provided as "commercial computer software" or "commercial computer software documentation" and, absent a written agreement to the contrary, the government's rights with respect to such Software are limited by the terms of this Agreement pursuant to FAR Paragraph 12.212(a) and/or DFARS Paragraph 227.7202-1(a), as applicable.

8. Export

Developer shall comply with all export and re-export restrictions, laws and regulations of the U.S. Commerce Department and other U.S. agencies and authorities.

9. General

9.1. Assignment. This Agreement shall be binding upon and inure to the benefit of the parties hereto, and their respective successors and permitted assigns. Developer shall not assign or otherwise transfer this Agreement or any rights or obligations hereunder, in whole or in part, whether by operation of law or otherwise, without Dialogic's prior written consent. Any purported transfer, assignment or delegation without such prior written consent will be null and void and of no force or effect.

9.2. Entire Agreement. This Agreement, together with any Special Terms, represents the entire agreement between the parties, and supersedes all prior agreements and understandings with respect to the matters covered by this Agreement. In the event of a conflict between this Agreement and the Special Terms, the Special Terms shall control. Developer agrees that it has not entered in this Agreement based on any representations other than those contained herein. No modification of or amendment to this Agreement, nor any waiver of any rights under this Agreement, shall be effective unless in writing signed by the party to be charged.

9.3. Governing Law. This Agreement shall in all respects be governed by the laws of the Commonwealth of Massachusetts, excluding: (i) its conflicts of laws principles; (ii) the United Nations Convention on Contracts for the International Sale of Goods; (iii) the 1974 Convention on the Limitation Period in the International Sale of Goods; and (iv) the Protocol amending the 1974 Convention, done at Vienna April 11, 1980.

9.4. Severability. If any of the provisions of this Agreement is held by a court of competent jurisdiction to be invalid under any applicable statute or rule of law, such provision shall, to that extent, be deemed omitted, and the remaining portions of this Agreement shall remain in full force and effect.

9.5. Waiver. The waiver of one breach or default or any delay in exercising any rights shall not constitute a waiver of any subsequent breach or default.

9.6. Notices. All notices permitted or required under this Agreement shall be in writing and shall be delivered in person or mailed by first class, registered or certified mail, postage prepaid, to the address of the party specified on the first page of this Agreement or such other address as either party may specify in writing. Such notice shall be deemed to have been given upon receipt.

9.7. Counterparts. This Agreement may be executed in counterparts, each of which shall be deemed an original and all of which together shall constitute one instrument.

9.8. Headings. The headings are for convenience and shall not be used to construe this Agreement.

9.9. No Agency; Independent Contractors. Nothing contained in this Agreement shall be deemed to imply or constitute that either party is the agent or representative of the other party, or that both parties are joint ventures or partners for any purpose.

EXHIBIT A

THIRD PARTY SOFTWARE LICENSES

The Software includes object code versions of the following third party software packages which are subject to the terms and conditions of the applicable license agreements found at the websites identified:

- ***ACE™ software and TAO™ software are copyrighted by Douglas C. Schmidt and his research group at Washington University, University of California, Irvine, and Vanderbilt University. Copyright (c) 1993-2003, all rights reserved. More information is available at:***

  http://www.cs.wustl.edu/~schmidt/ACE.html

- ***wxWindows software is copyrighted by Julian Smart, Robert Roebling et al. Copyright (c) 1998 Julian Smart, Robert Roebling et al. More information is available at:***

  http://www.wxWindows.org

- ***Portions of the Software are licensed under the Apache License, Version 2.0 (the "Apache License"); you may not use this file except in compliance with the Apache License. You may obtain a copy of the Apache License at:***

  http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software distributed under the Apache License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the Apache License for the specific language governing permissions and limitations under the Apache License.

EXHIBIT B

Limited Warranty to End Users

Developer may include the following warranty terms in the End User License Agreement:

[10].1. Limited Warranty

a. Dialogic warrants to End User that for a period of ninety (90) days from delivery of the Software to End User (the "Software Warranty Period"), the Software will substantially perform in accordance with its documentation. Dialogic's sole liability and End User's exclusive remedy under this limited warranty shall be (i) the repair or replacement of the non-conforming Software or, (ii) at Dialogic's option, in the case of Software licensed for use with Dialogic Hardware, the refund of the price paid by End User for the Dialogic Hardware or, in the case of software licensed and distributed as host-based media processing software, the refund of the price paid by End User for the License Key.

b. The remedies set forth in Section ***[10]***.1(a) are available only if Dialogic is promptly notified in writing, within the applicable warranty period, upon discovery of the non-conformity by End User, and Dialogic's examination of the Software and/or Dialogic Hardware discloses that such non-conformity exists. This limited warranty shall not apply if the Software or Dialogic Hardware has been: (i) altered or modified; (ii) subjected to negligence, computer or electrical malfunction; or (iii) used, adjusted, installed or operated other than in accordance with instructions furnished by Dialogic or with an application or in an environment other than that intended or recommended by Dialogic.

[10].2. Disclaimer of Any Other Warranties. EXCEPT FOR THE EXPRESS, LIMITED WARRANTY PROVIDED IN SECTION [10].1, DIALOGIC MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THE SOFTWARE, AND DIALOGIC DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. DIALOGIC DOES NOT WARRANT THAT THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT IT WILL SATISFY END USER'S REQUIREMENTS.

Contents

This chapter provides a series of steps that you can perform to quickly set up and test your installation of the Computer Telephony and software.

Chapter 2 – Installing the Developer Kit on Red Hat Linux 9

This chapter describes how one can install Dialogic® Brooktrout® software on Red Hat Linux operating systems.

Chapter 3 – Installing the Developer Kit on Solaris 27

This chapter describes how one can install Dialogic® Brooktrout® software on Solaris operating systems.

This chapter describes how one can configure the Dialogic® Brooktrout® software for your Dialogic® Brooktrout® boards.

This chapter contains tables that show the directory structure used for all installed binary, firmware, and sample source files, both platform-independent and platform-specific for all operating system platforms.

Chapter 8 – Dialogic® Brooktrout® AccuCall and the AccuCall Wizard. 157

This chapter describes how one can use the Dialogic® Brooktrout® AccuCall Wizard to train your board to recognize tones from your PBX and how to create and edit Programmable Call Progress Monitoring (PCPM) tables for use with different Private Branch Exchanges (PBXs).

Chapter 9 – Using the Dialogic® Brooktrout® TECUpdate Utility 237

This chapter describes how one can use the Dialogic® Brooktrout® Technology Expansion Capability (TECUpdate) utility to display and upgrade the configured feature set on an installed board.

Chapter 10 – Downloading Firmware 245

This chapter describes how one can download the firmware to the board.

Appendix A – Compliance Information 253

This appendix describes how the Dialogic® Brooktrout® boards conform to compliance standards.

This appendix describes how one can create a custom installation program for manual configuration of the driver.

Preface

About this Manual

The Dialogic® Brooktrout® ***TR1000 Series SDK: Installation and Configuration Guide*** explains how to install the software (firmware, Bfv API, and driver for the Dialogic® Brooktrout® TR1000/TR1034/SR140/TruFax® boards) on your host system. It also describes how to configure the driver, configure call control, and download the firmware to a hardware module.

- Chapter 1 provides a quick start for users experienced with API programming in general and the Bfv API in particular.
- Chapter 2 explains how to install the Dialogic® Brooktrout® software and the device driver on a Red Hat Linux operating system.
- Chapter 3 explains how to install the Dialogic® Brooktrout® software and the device driver on the Solaris operating system.
- Chapter 4 explains how to install the Dialogic® Brooktrout® software on a Windows® operating system.
- Chapter 5 describes how to install the Dialogic® Brooktrout® device driver on Windows® operating systems.
- Chapter 6 describes how to configure your software and hardware on Windows® systems.
- Chapter 7 lists the directories where files are installed.

- Chapter 8 describes how to use the Dialogic® Brooktrout® AccuCall Wizard to train your board to recognize tones from your PBX and how to create and edit PCPM tables for use with different Private Branch Exchanges (PBXs).
- Chapter 9 describes how to use the Dialogic® Brooktrout® TECUpdate Utility.
- Chapter 10 describes the firmware files, how to load them, and how to update the boot ROM.
- Appendix A describes compliance with North American and European standards that includes general telecommunications and safety.
- Appendix B describes how to create a custom installation program for the manual configuration of the driver.
- Appendix C describes how to use the driver reinitialization utility.
- A glossary provides definitions for some of the terms used in the manual.

A copy of this manual in Adobe Acrobat PDF format is included in the ***Documents*** directory on the TR1000 Series SDK CD-ROM.

Refer to Dialogic® Brooktrout® SR140 Fax Software User Guide for instructions to activate the software using the License Manager.

Operating System Support

See the latest Dialogic® Brooktrout® ***TR1000 Series SDK: Release Notes*** for the supported operating systems. A copy of the Release Notes is found in the ***Documents*** directory on the TR1000/TR1034/SR140/TruFax® Developer Kit CD-ROM.

Compile Compatibility

Applications compiled with either SDK 3.2, SDK 3.3, SDK 4.0, SDK 4.1, or 5.0 work without recompiling in SDK 5.2.0 as long as the application called the BT_API_SET_VER macro. See the ***Bfv API Reference Manual, Volume*** 1.

You might also require one of the following:

- Windows® libraries, see ***Special Considerations on page 80***

- Linux libraries, see ***Special Considerations on page 18***
- Solaris libraries, see ***Special Considerations on page 37***

Manual Conventions

This manual uses the following conventions:

- ***Italics*** denote file names, directory names, and program names within the general text.
- The **`Courier`** font in bold indicates a command sequence entered by the user at the system prompt, for example:

  `cd /boston/bfv.api`
- The `Courier` font not bolded indicates system output, for example:

  `C:>Files installed.`
- The `Courier` font also denotes programming code, such as C and C++. Programming code appears in program examples.
- ***Bold*** indicates names of variables, Bfv API functions, dialog boxes, dialog box controls, windows, and menu items.
- Square brackets [ ] indicate that the information to be typed is optional.
- Angle brackets < > indicate that you must supply a value with the parameter.

The Caution icon is used to indicate an action that could cause harm to the software or hardware.

The Warning icon is used to indicate an action that could cause harm to the user.

Related Documents

Hardware Documents

These installation cards come with their respective boards:

- ***Installing the Dialogic® Brooktrout® TR1000 Digital PCI Board (Single/Dual Span)*** explains how to install a TR1000 digital PCI board with an H.100 interface.
- ***Installing the Dialogic® Brooktrout® TR1000 Series Digital Board (Multiple Spans)*** explains how to install a TR1000 or TR1034 digital PCI board with multiple spans.
- ***Installing the Dialogic® Brooktrout® TR1000 Analog PCI Board*** explains how to install a TR1000 analog PCI board.
- ***Installing the Dialogic® Brooktrout® TR1000 Analog PCI Board with H.100 Support*** explains how to install a TR1000 analog PCI board equipped with an H.100 module.
- ***Installing the Dialogic® Brooktrout® TR1000 BRI PCI Board*** explains how to install a TR1000 BRI PCI board.
- ***Installing the Dialogic® Brooktrout® TR1000 BRI PCI Board with H.100 Support*** explains how to install a TR1000 BRI PCI board equipped with an H.100 module.
- ***Installing the Dialogic® Brooktrout® TR1034 Digital/Ethernet PCI Board (Single Span)*** explains how to install a Dialogic® Brooktrout® TR1034 digital PCI board with an H.100 interface.
- ***Installing the Dialogic® Brooktrout® TR1034 Analog PCI Board*** explains how to install a Dialogic® Brooktrout® TR1034 analog PCI board.
- ***Installing the Dialogic® Brooktrout® TR1034 Analog LS/DID Combination PCI Board*** explains how to install the board and connect telephone service for both Analog Loop Start (ALS) and Direct Inward Dialing (DID).
- ***Installing the Dialogic® Brooktrout® TR1034 BRI PCI Board*** explains how to install a Dialogic® Brooktrout® TR1034 BRI PCI board.
- ***Installing the Dialogic® Brooktrout® TruFax® 100-R and 200-R Analog PCI Boards*** explains how to install a Dialogic® Brooktrout® TruFax® analog PCI board.

- ***Installing the Dialogic® Brooktrout® TruFax® BRI PCI Boards*** explains how to install a Dialogic® Brooktrout® TruFax® BRI PCI board.

Software Documents

For product information, white papers, FAQs, and more, access the Dialogic web site at ***www.dialogic.com***.

These volumes come with the SDK:

- The ***Dialogic® Brooktrout® TR1000 Series SDK Developer Guide*** describes the Bfv API and gives information about Call Transfer, Automatic Speech Recognition, the Dialogic® Brooktrout® Audio Provider, BSMI, and how to package Dialogic® Brooktrout® software for your customers.
- ***Volume 1, Administration, Management, and Configuration, Bfv API Reference Manual*** provides information about the following Bfv API components:
  - Administration and Initialization functions
  - Firmware, Configuration, Status and Monitoring functions
  - Debugging, Error Handling and Return Values
  - Miscellaneous functions
- ***Volume 2, Bfv-Level Call Control and Call Switching, Bfv API Reference Manual*** provides information about the following Bfv API components:
  - Bfv API-level Call Control functions
  - Dialing Database functions
  - Call Control data structures and macros
- ***Volume 3, Media Processing Bfv API Reference Manual*** provides information about the following Bfv API components:
  - Signal Generation and Detection functions
  - Voice Play and Record functions
  - Infopkt file functions
  - Audio Conferencing functions
- ***Volume 4, Fax Processing, Bfv API Reference Manual*** provides information about the following Bfv API components:
  - Fax functions and macros
  - TIFF-F files functions and macros

- ***Volume 5, BSMI-Level Call Control and Call Switching, Bfv API Reference Manual*** provides information about the following BOSTON Simple Message Interface (BSMI) Bfv API components:
    - BSMI-level Call Control functions
    - Message structures
    - R2 Signaling Protocol messages
    - Local Exchange Carrier (LEC) Protocol messages
    - Host to Module and Module to Host messages
    - B-Channel and D-Channel Maintenance procedures
- ***Volume 6, Appendices, Bfv API Reference Manual*** is a grouping of appendices that relate to the reference material in Volumes 1 through 5, including:
    - Configuration files
    - Bfv API structures
    - Hangup codes, Cause codes, Infopkt parameters
    - Country-specific dialing requirements
    - Call Progress notes
    - Deprecated and unsupported functionality

Getting Technical Support

Dialogic provides technical support for customers who have purchased hardware or software products from Dialogic. If you purchased products from a reseller, please contact that reseller for technical support.

To obtain technical support, please use the web site below:

www.dialogic.com/support/

1 - Quick Start Steps

This chapter provides a series of steps that you can perform to quickly set up and test your installation of the Computer Telephony and software.

Because the installation and configuration procedures can be quite different for Windows® systems from those for Linux and Solaris systems, this chapter gives separate quick start instruction for Windows® systems.

This chapter includes the following:

Determining the Board Module Number

You need to know the module numbers (IDs) and associated channel numbers of each Dialogic® Brooktrout® board (for example: TR1000, TR1034, TruFax®, or SR140 virtual module) in your system for a variety of tasks, including:

- Firmware download
- Telephony and network configuration
- Using some utilities (see the chapter in the developer guide on sample applications and utilities)

Note: The Dialogic® Brooktrout® TR1000 and Dialogic® Brooktrout® TR1034 are also herein referred to as the TR1000 and TR1034 respectively. The Dialogic® Brooktrout® SR140 is also herein referred to as the SR140. The Dialogic® Brooktrout® TruFax® Fax Board is also referred to as the TruFax® Fax Board.

A hardware module is a unit that communicates (using the BOSTON software) on a Dialogic® Brooktrout® board. In general, each module corresponds to a control processor on the board. Most Dialogic® Brooktrout® boards have one module. The driver assigns each system with its own unique module number. Module numbers are normally referred to and displayed in hex.

Dialogic® Brooktrout® PCI boards contain a rotary switch that allows selection of a module number in a range from 2 through F.

To find the module numbers of the Dialogic® Brooktrout® boards in your system, run the ***modinfo*** program:

```
modinfo
```

The ***modinfo*** program provides the number of channels on each module. See the chapter on sample applications and utilities in the developer guide for more information.

The driver assigns other module numbers in addition to those for hardware modules. These can often be seen when running ***modinfo,*** for example.

- Module 1 represents the driver.
- Module FE is shared by all applications.
- Module 0x41 is the default for Dialogic® Brooktrout® SR140 (virtual modules).

Host Based Fax (Dialogic® Brooktrout® SR140) Modules

Host Based Fax (HBF) is a software-based virtual module (Dialogic® Brooktrout® SR140) that allows you to use T.38 fax without installing Computer Telephony. It is supported on Windows® systems only and uses IP-enabled call control through SIP and H.323 stacks.

If you have SR140 modules, you must install the TR1000/TR1034/SR140/TruFax® Developer Kit SDK software first. Then you can turn on Dialogic® Brooktrout® SR140 modules using the Dialogic® Brooktrout® License Manager. For more information, see:

- ***Chapter , Installing the Developer Kit on Windows® on page 45***
- ***Chapter 10, Activating Brooktrout Products on page 277***

Quick Start for Windows® Systems

If you have SR140 modules (software only), see ***Host Based Fax (Dialogic® Brooktrout® SR140) Modules on page 3*** for setup information, otherwise continue with Step 1.

Step 1: Install Computer Telephony and Software

➢ **Do the following to install the Computer Telephony and software:**

1. Before you install a Dialogic® Brooktrout® board, install the SDK by inserting the TR1000/TR1034/SR140/TruFax® Developer Kit SDK from the CD-ROM drive. Then follow the directions in ***Chapter , Installing the Developer Kit on Windows® on page 45***.

   This procedure places the contents of the CD into a directory on your hard drive. The directory is either one you specify or the default (***\Brooktrout\Boston***).

   See ***Chapter , Directory Structure on page 147*** for the list of directories and their contents in this structure.

2. Power down the computer and install the board in a PCI slot.

   For installation instructions, see the hardware installation card that came with your board.

   Although you can install the hardware before you install the software, Dialogic recommends that you install the software first.

3. Power up the computer and log on.

   - When the status LED on the board is flashing yellow, the board is ready to have the firmware downloaded (see ***Step 2: Configuration Tool and Firmware Starts on page 5***).
   - When the system sees the TR1000/TR1034/TruFax® board, install the driver on a Windows® operating system (see ***Installing the Plug-and-Play Driver on page 59***).

Step 2: Configuration Tool and Firmware Starts

➢ **To run the configuration tool and download firmware, use these steps:**

1. Run the Dialogic® Brooktrout® Configuration Tool ***configtool.exe*** to configure the driver and the call control parameters. See ***Using the Dialogic® Brooktrout® Configuration Tool on Windows Systems on page 91***.

   The Configuration Tool also starts the driver and downloads the firmware to the module.

Note: For the Dialogic® Brooktrout® Configuration Tool to work with Windows® 2000, you must install Internet Explorer 6.0 or higher.

2. Reboot the system after running ***configtool.exe*** to have the driver configuration take effect.

Step 3: Compile and Run the Sample Application

To test your board and its setup, you can run the voice sample application, found in the ***Boston\bfv.api\app.src*** directory. However, before you can run the sample application, you must compile it. To compile sample applications, follow the instructions in the developer guide (see the chapter about ***Sample Applications and Utilities***).

Next Steps

After you have verified that your board is functioning properly and can record and play speech, you can run your existing application, test your new application, or create a new application.

For more information, see these chapters in the developer guide:

- ***Introduction to the Bfv API*** describes the Bfv API
- ***Debugging*** describes how to debug your applications
- ***Sample Applications and Utilities*** describes how to compile and use the sample applications

Quick Start for Linux and Solaris Systems

Note: The SR140 is not supported on Linux and Solaris operating systems.

Step 1: Install Computer Telephony and Software

➢ **Do the following to install Computer Telephony and software:**

1. Install the Dialogic® Brooktrout® board in a PCI slot in your computer chassis. For installation instructions, see the hardware installation card that came with your board.

   When the status LED on the board is flashing yellow, it is ready to have the firmware downloaded. See Step 2.

2. Insert the SDK CD into a CD drive and follow the directions in the applicable chapter for your operating system (*Chapter , Installing the Developer Kit on Red Hat Linux on page 9* or *Chapter , Installing the Developer Kit on Solaris on page 27*).

   This procedure places the contents of the CD into a directory on your hard drive. The directory can be one you specify or the default, ***/usr/sys/brooktrout/boston***. See *Chapter , Directory Structure on page 147* for the list of directories and their contents in the ***/boston*** directory structure.

3. Start the driver. Linux and Solaris, if you want, can start the driver, but you can wait until the software is installed before you start the driver. See the appropriate chapter that applies to your operating system.

Step 2: Select and Download the Firmware Files

Select the Firmware Files

All Dialogic® Brooktrout® boards need certain firmware files including:

- The control processor firmware called ***cp.bin.***
- The correct DSP firmware file for your board and product. See ***Chapter , Downloading Firmware on page 245*** for the names of the DSP firmware files.

Download the Firmware

See ***Manually Downloading the Firmware on page 249*** for the instructions for downloading the firmware.

Step 3: Configure the Call Control Parameters

Dialogic recommends that you use the call control configuration file named ***callctrl.cfg*** to configure call control parameters. See ***Using a Call Control Configuration File on page 145*** for more information.

Step 4: Compile and Run the Sample Application

To test your board and its setup, you can run the voice sample application, found in the ***bfv.api/app.src*** directory. However, before you can run the sample application, you must compile it. To compile sample applications, follow the instructions in the chapter on compiling the sample applications in the developer guide.

Next Steps

After you have verified that your board is functioning properly and can record and play speech, you can run your existing application, test your new application, or create a new application.

For more information, see these chapters in the developer guide:

- ***Introduction to the Bfv API*** describes the Dialogic® Brooktrout® Bfv API.
- ***Debugging*** describes how to debug your applications.
- ***Sample Applications and Utilities*** describes how to compile and use the sample applications.

2 - Installing the Developer Kit on Red Hat Linux

This chapter describes how one can install Dialogic® Brooktrout® software on Red Hat Linux operating systems.

This chapter contains the following:

Installation Steps

➢ **To install the Dialogic® Brooktrout® software on Linux, follow the steps below:**

When running driver installation programs, scripts, or functions, you must be logged in as `root` or have administrative privileges. Dialogic supports only the standard GNU C compiler that ships with Linux.

1. Install the firmware, driver, and Bfv API software onto your computer.

   If you are reinstalling or upgrading software from a previous SDK, you must first remove the previously installed software before installing this software.

   See ***Removing the Software on page 13***.

2. Compile the sample applications and utilities (optional). See the chapter on sample applications and utilities in the developer guide.

3. Run the `dinstall` program to manually configure the driver (optional).

   See ***Configuring the Driver Manually on page 14***.

4. Download the firmware, optionally configure the telephony parameters, see:

   - ***Chapter , Downloading Firmware on page 245***
   - ***Using a Call Control Configuration File on page 145***).

Installing the Software

The TR1000/TR1034/SR140/TruFax® Developer Kit CD-ROM comes with an installation program that uses ***rpm*** to install the files on your hard drive. To use the program, follow the directions below. You must be logged in as `root` or `su` to install/uninstall the software and configure the driver.

➢ **Use the following procedure to install software files:**

1. Mount the installation CD-ROM:

   `mount -t iso9660 -r /dev/cdrom /mnt/cdrom`

2. Enter the command line to go to the ***Linux*** directory on the CD-ROM drive:

   `cd /mnt/cdrom/Software/Linux/Red Hat`

3. Run the setup program by entering the following at a command prompt:

   `sh ./setup.sh`

   The program displays a message if it does not detect an Intel microprocessor and terminates. If the processor is Intel, the program then displays a copyright notice and the following information:

```
Welcome to the Brooktrout Setup Program.
Program to install Boston API/Driver and Firmware Packages
WARNING:  This program is protected by copyright law and international
treaties.
Unauthorized reproduction or distribution of this program, or any portion
thereof, may result in severe civil and criminal penalties, and will be
prosecuted to the maximum extent possible under law.
```

The program issues the following message only if you have the driver called `bfax`, that runs on the TR114 board:

```
The Brooktrout Driver package called "bfax" has been detected on your system. It
is OK to have both "BRKTBOSsetup" and "bfax" installed on your system. Since these
packages share common files, you may encounter errors during package removal.
```

   The program then displays:

```
Press ENTER to continue or ’q’ then ENTER to quit
setup.
```

4. The program then displays:

```
Default install directory is /usr/sys/brooktrout

Please enter desired install directory or just press
ENTER to use default directory.
```

5. After you enter an install directory or accept the default directory location, the program prompts you to identify files to install:

```
1) Boston API/Driver Package.
2) Firmware Package.
3) Boston API/Driver and Firmware Packages.
0) Exit Installation.
```

```
Please enter the number corresponding to the files you
wish to install?
```

6. Type the number of your choice.

   If the packages you selected have been installed, the program gives you as message to that effect, such as:

```
The Boston API/Driver Package was installed
previously.
```

   Otherwise, the program gives you a message such as:

```
Running RPM to install the Boston API/Driver Package
```

7. The program then asks:

```
Would you like to configure the driver at this time
[y or n]?
```

If you choose Boston API/Driver and Firmware Packages and the default directory, the directory ***/usr/sys/Brooktrout*** contains the directories and files shown in ***Chapter , Directory Structure on page 147***.

After installing the software, you must download the firmware. See ***Chapter , Downloading Firmware on page 245***.

Removing the Software

When you install the software on Linux, you can choose one or two packages. If you choose to install the "Bfv Binary files," ***rpm*** installs a package called ***BRKTBOSsetup***. If you choose "Firmware files," ***rpm*** installs a package called ***BRKTBOSsetup-fw***.

If you choose to install "Bfv Binary files and firmware files," then both of the above packages are installed.

Each package must be removed with a separate command.

➢ **To remove all files and directories created during the installation, enter:**

```
rpm -e BRKTBOSsetup-fw
rpm -e BRKTBOSsetup
```

These commands remove only the files and directories that the installation procedure placed on the system.

Starting or Stopping the Driver

You must start or stop the driver as shown below. You can also add the command lines to system startup scripts, if needed.

➢ **To start the driver, enter:**

```
/etc/startbost
```

➢ **To stop or shut down the driver, enter:**

```
/etc/startbost -r
```

Reinitializing the Driver

You can reinitialize the driver to its starting state in terms of channels assigned to modules and module number assignments using the ***driver_reinit.exe*** utility located in the ***boston/driver/linux/user*** directory (see ***Reinitializing the Device Driver on page 285***).

Configuring the Driver Manually

To configure the driver manually, change to the ***/usr/sys/brooktrout/boston/driver/linux/install*** directory and run the ***dinstall*** program.

The configuration dialog is as follows:

```
Installing Brooktrout Boston Device Driver Version 5.2.0
Maximum number of PCI/cPCI hardware modules (default 16):
Physical buffer size (default 32768):
Application buffer size (default 10240):
Machine ID, in hex (default 1):
Do initial reset (default 1):
History enable (default 0):
```

If you enter ***1***, ***dinstall*** asks the following questions:

```
History size (default 1024000):
Number of physical channel histories (default 0):
Number of application channel histories (default 0):
Configure advanced parameters (n)?
```

If you enter ***y***, ***install*** asks the following questions:

```
Number of I2O frames per module (default 16):
Driver flow control interval (default 250):
Internal memory allocation minimum (default 152):
Internal memory allocation quanta (default 128):
To reserve module numbers, enter mod_id (serial number)
and mod_num, in hex. Mod_num will default as indicated:
mod_id mod_num[F0]:
Configure test parameters (n)?
(If you enter y, three test options are listed)
Test1:
Test2:
Test3:
```

```
Configuring Boston driver:
   Phys buf size 32768
   App buffer size 10240
   Machine ID 1
   History size 1024000
   Num phys hist bufs 0
   Num apl hist bufs 0
```

The maximum number of PCI hardware modules can be left at its default. However, if fewer modules are in use, to conserve memory, this value can be set to a smaller number. Each hardware module uses approximately 60K.

History Logging

When the Boston driver is installed, history logging is disabled by default. When you manually install the driver using ***dinstall***, the default is set to disable history logging (you can choose to override this option). History logging is a powerful debugging tool, especially when working with Technical Support. However, it does consume a significant amount of CPU time when running very high call loads. Therefore, leave history logging off for systems that have high call loads and are CPU-bound. For low-load systems or when debugging, enable history logging.

Leave the number of physical and application channel histories at 0 unless you are directed to change this by Technical Support. If you change the physical or application buffer size, do not change the value to one smaller than the default.

If you specify a number for the physical channel histories, ***dinstall*** asks if you want to restrict the physical channels to administrative channel 1.

```
Restrict physical channels to admin channel 1 (default 1):
```

Choose the default (enable).

Advanced Configuration Parameters

If you select advanced configuration parameters, the software prompts you for the number of I2O frames, driver flow control interval, internal memory allocation parameters, module reservations, and test parameters. Do not configure advanced parameters except under the advice of Dialogic Technical Support.

Linux Kernel Versions

Different versions of Linux have different kernel version numbers, and patches to Linux can also change the kernel version number. The binary distribution comes with driver support for a number of kernel versions. Dialogic® Brooktrout® drivers will be installed for all the installed kernel versions that are supported. If the currently installed kernel version is not supported, driver installation aborts.

The driver is configured and ready to be started.

See ***Starting or Stopping the Driver on page 13*** and ***Recompiling on Linux Platforms on page 24***.

Removing the Manually Configured Driver

➢ **To uninstall the driver, use these steps:**

1. Change to the following directory:
   /usr/sys/brooktrout/boston/driver/linux/install
2. Enter the following at the command prompt:

   `./dinstall -r.`

Using Bostsrv Under Linux

Boston Host Service (Bostsrv) is used to initialize modules. Upon startup it detects Dialogic® Brooktrout® modules, loads FW and initializes the call control. You must use Boston Host Service to run a multi-process application or to use SR140 Host Based Fax (HBF) modules. Boston Host Service is optional for other Dialogic® Brooktrout® modules.

Bostsrv is installed and started automatically by the Configuration Tool when you click the Apply button. The command line process is not required.

When you use the Boston Host Service (Bostsrv) with a Linux system the following syntax applies:

- Go to following directory:

```
cd /usr/sys/brooktrout/boston/bfv.api/linux/bin/7/Bostsrv
```

- Use any of the following command line options:

Option	Meaning
-c <file>	Full path to the ***btcall.cfg*** file and name of the ***btcall.cfg*** file
-D	Run as daemon (in background)
-f <path>	Full path to firmware
-k	Stops the daemon
-stop	Stops the daemon
-query	Queries whether the server is running
-debug	Console app/debugging
-v	Enable Bfv API debug mode

Note: **-c** and **-f** are required.

For example:

```
bostsrv -c /usr/sys/brooktrout/boston/config/btcall.cfg
-f /usr/sys/brooktrout/boston/fw
```

Root login for Unix

When using Unix, you must log in as root to run Bostsrv from the command line in debug mode. Bostsrv creates a process id file in the root `owned/var/run` directory. This prevents multiple copies from running at the same time. It also supports the kill, stop, and query operations.

Special Considerations

When you install the Software on a Red Hat Linux operating system, consider the following:

PCI Considerations

Dialogic currently only provides PCI boards in the TR1000 Series and TruFax® models. When configuring the device driver, you must select 0 when asked for the number of ISA boards.

The operating system and/or system BIOS assigns base addresses and interrupt numbers to the PCI boards, which can change when the system is rebooted, especially if devices are added or moved.

If multiple PCI boards are in use, it is often not apparent which boards correspond to which module numbers or contain which ordinal channel values. See ***Determining the Board Module Number on page 2*** for more information about modules and module number assignment.

Bfv API Libraries and Makefiles

The Dialogic® Brooktrout® Bfv API libraries are supplied in several forms for the appropriate Linux versions. The primary libraries are shared (.so) libraries. In addition, there are static (.a) libraries; their use is deprecated and are supplied for compatibility only.

To run an application compiled with the ***.so*** libraries, the operating system needs to find the ***.so*** libraries at runtime. The installation procedure automatically installs these libraries into a known system location: ***/usr/lib***. If you are performing installation manually, you can either copy the libraries there, or add the directory locations containing the ***.so*** libraries to the environment variable LD_LIBRARY_PATH.

Follow the makefiles supplied in the ***app.src*** or ***bapp.src*** directory for all applications. In particular, since all the supplied libraries were compiled as multithreaded, compile all your applications this way. See also, ***Reviewing Compiler and Linker Options on page 23***.

Writing Multithreaded Programs

The Bfv API supports multithreaded programs using the ***pthreads*** package. When developing a multithreaded program, be sure to follow the programming guidelines described in your system documentation.

When new threads are created, often a stack size can be specified. The stack size must be sufficiently large to store all local variables used both by the application and the API. If the stack size is not large enough, unpredictable behavior can result.

Dialogic usually finds a stack size of 64K to be sufficient but, depending on the application's stack allocation demands, more memory might be required.

Warning and Error Messages

If the driver has an error or warning to report (often related to driver startup problems), the report always appears on the system console, if such a device exists for that platform. Additionally, a limited amount of messages are sometimes available if you use the `crash` command and select the `panic` option. These messages are logged to the ***/var/log/messages*** text file.

Determining Installation Parameters

Any time after you install the driver (but not removed) you can determine what installation parameters were used by looking at the ***/etc/startbost*** file.

Command Line Configuration for dinstall

When configuring the device driver using ***dinstall***, you can specify the command line option `-c` to cause the ***dinstall*** program to not prompt you for any parameters. Instead, the ***dinstall*** program uses the responses that you supply following the `-c` option on the command line.

If you use the `-c` option, you must include all the parameters described below, in order, regardless of whether or not the parameter has meaning on the system in question. If a parameter does not apply, it is ignored. The parameters are not checked for validity.

Note: Prompting for removal of old driver versions is not performed.

When configuring the driver using ***dinstall***, you must include the following parameters:

Parameter	Defaults
Reserved	0
Reserved	-1
Max num PCI/cPCI hw modules	16
Physical buffer size	32768
Application buffer size	10240

Reserved	0
Machine ID	1
Reset option (1 or 0)	1
History Enable	0
History Size	1024000
Num physical histories	0
Num app histories	0
Restrict phys channels (1 or 0)	1
Number of I2O frames	*0
Flow control interval	*0
Memory alloc minimum	*0
Memory alloc quanta	*0
Reserved	0
Test parameter 1	*0
Test parameter 2	*0
Test parameter 3	*0
Reserved module info module id, module number, 0 to terminate	*0
Reserved	0

The advanced parameters, marked with * in the list above, can always be safely set to 0. This tells the driver to use the default values.

There are currently no parameters used with `-c` when removing the driver, but the option might still be given on the command line for future use.

For example, to configure the driver for 0 ISA hardware modules and up to 16 PCI hardware modules, use the following settings:

- physical buffer size 32768
- application buffer size 10240
- machine ID 1
- reset on
- history enabled
- history size 1024000
- physical histories 0

- application histories 0
- restrict physical channels
- all default advanced parameters

Enter the following at the command line:

```
./dinstall -c 0 -1 16 32768 10240 0 1 1 1 1024000 0 0 1 0 0 0 0 0 0 0 0 0 0
```

Driver File Locations

When using ***dinstall*** to configure the device driver, you need not have all files from the distribution. However, the locations of files used by the installation procedure relative to the directory from which ***dinstall*** is being run must be the same as that in the normal distribution.

Assuming a directory named ***install*** from which ***dinstall*** is being run, the following files must be present:

install/dinstall

install/dinstlib

kernel/kvers/*

These files and directories appear in the standard distribution under the directory ***driver/linux***.

The ***kvers*** directory contains subdirectories named after each kernel version for which binaries are available. The software supports the standard uni and multiprocessor releases, as well as alternate releases. See the Release Notes for the supported kernel versions.

Reviewing Compiler and Linker Options

The makefiles contained in the ***app.src*** and ***bapp.src*** directories provide a number of command line options to the compiler and linker. Using these options, the compiler and linker produce object files and executables compatible with the Bfv API library.

Dialogic strongly recommends that you use these options for all your Dialogic® Brooktrout® Bfv API applications. But for those who want to change the set of options the compiler and linker use, the following is a list of the nonstandard options and their functions. The options used for compilation are marked with (***C***) and those for linking are marked with (***L***).

`-DBFAX_THREADS`	(***C***) Specifies a multithreaded program to the API.
`-lpthreads`	(***L***) Multithreaded library.

Recompiling on Linux Platforms

This section provides instructions for recompiling the Boston driver to support new kernel patches.

Use the instructions below to recompile the Boston driver on supported Linux platforms so that the driver can operate with any kernel patch for supported Linux versions. Dialogic only supports official kernel patches as released by Red Hat. After you follow the procedure, the driver supports only the exact version of the kernel currently running on your system, including architecture and variant.

Note: This feature only provides support for the Boston driver, the kernel mode code. Other parts of the SDK (the user mode code) might also need rebuilding and this feature will not help in these situations.

Supported versions include:

- Red Hat Enterprise Linux ES/AS 3.0 and 4.0

The fully precompiled installable driver binaries reside under the ***driver/linux/kernel/kvers*** directory. This directory contains subdirectories corresponding to each kernel version, variant, and architecture, with a driver binary in each (named ***boston.o*** or ***boston.ko***).

For each Linux OS version supported, the only precompiled driver supplied supports the original kernel that shipped with that version of Linux.

Each of these directories also contains a file named ***bostbase.a***, which is a library containing precompiled object files compiled for that same kernel version, variant, and architecture.

The driver/linux/kernel directory contains files named ***kerndep.c***, ***kerndep.h***, and ***makefile.kerndep***.

Before building a Dialogic® Brooktrout® driver for a patch version, install the kernel source, the compiler, and other standard development tools on the system.

➢ **To build a driver for a kernel patch version, enter the following at the command prompt:**

```
make -f makefile.kerndep
```

This command performs several steps.

- Determines what the base Red Hat release is that the booted kernel is based on, what the kernel version is that corresponds to the base release, and what the variant and architecture are.
- Compiles the source file ***kerndep.c*** on the current booted kernel setup.
- Links the resulting object file with the ***bostbase.a*** file from the directory corresponding to the base kernel version for the current variant and architecture.
- Puts the resulting driver binary into an appropriate ***kvers*** subdirectory for the actual kernel version in use.

After compiling the driver, use the standard manual ***dinstall*** script provided on the CD to manually configure the driver.

If the kernel source is not installed in a standard location, use the optional KERNEL_SOURCE=<dir> command line option to specify the kernel source location to the ***make*** utility.

For Red Hat Linux releases ES/AS 3.0 and earlier:

The kernel source is the package whose name is of the form ***kernel-source-<version>***, in the file ***kernel-source-<version>.i386.rpm***. This package is automatically installed if you tell the Linux installation program to install everything.

For Red Hat Linux releases ES/AS 4.0 and later:

The kernel source itself is not required; instead, a ***development kit*** is required. This kit is in the package whose name is of the form ***kernel-devel-<version>***, in the file ***kernel-devel-<version>.i686.rpm*** or ***kernel-smp-devel-<version>.i686.rpm***. This kit is automatically installed if you tell the Linux installation program to install everything. In this case, KERNEL_SOURCE should point to the appropriate subdirectory of ***/usr/src/kernels*** which would normally be one of ***<version>-i686*** or ***<version>-smp-i686***.

3 - Installing the Developer Kit on Solaris

This chapter describes how one can install Dialogic® Brooktrout® software on Solaris operating systems.

Note: The SR140 is not supported on Solaris operating systems.

This chapter has the following sections:

- *Installation Steps on page 28*
- *Configuring the Driver Manually on page 33*
- *Special Considerations on page 37*
- *Reviewing Compiler and Linker Options on page 43*

Installation Steps

To install the software on Solaris, follow the steps below. When running driver installation programs, scripts, or functions, you must be logged in as root or have administrative privileges.

The software is for either SPARC® or Intel x86. Dialogic supports only official development environments from Sun, such as Sun Workshop, Sun Pro, and Forte. In particular, GNU development environments are not supported.

Note: The SR140 is not supported on Solaris operating systems.

➢ ***To install hardware and software, use the following steps:***

1. Install the firmware, driver, and Bfv API software onto your computer.

   If you are reinstalling or upgrading the software, you must first remove the previously installed software before installing this software.

   See ***Removing the Software on page 31***.

2. Compile the sample applications and utilities (optional).
   See in the developer guide the chapter on compiling the sample applications.

3. Run the ***dinstall*** program to configure the driver (optional).

   See ***Configuring the Driver Manually on page 33***.

4. Download the firmware and configure the call control parameters, see:

   - ***Chapter , Downloading Firmware on page 245***
   - ***Using a Call Control Configuration File on page 145***

Installing the Software

The SDK CD-ROM comes with an installation program that uses ***pkgadd*** to install the files on your hard drive. To use the program, follow the directions below. You must install/uninstall the software and configure the driver from the ***root*** directory.

➢ ***To install files on your hard drive, do the following:***

1. Insert the installation CD-ROM.

   The CD-ROM splash screen should automatically display in the File Manager window. If it does not, check with your Solaris documentation for instructions for displaying the CD-ROM.

2. Traverse to the ***Solaris*** directory on the CD-ROM by entering this command line:

   cd /cdrom/<volume label>/Software/Solaris

   Where <volume label> is the label on the CD (for example, tr1000v31).

3. Go to the subdirectory with the Solaris files for x86 or SPARC. Enter either:

   cd "Solaris x86

   cd "Solaris SPARC"

4. Run the setup program by entering the following at the command prompt:

   sh ./setup.sh

   The program begins with the following messages:

```
The following packages are available
1. BOSsetup TR1000 Series 5.0.0 SDK (SPARC) 5.0.0
Select packages you wish to process (or ‘all’ to
process all packages) (default: all) [?,??,q]:
```

5. Select **1.**

   The program displays the following information:

```
WARNING:  This program is protected by copyright law and international treaties.
Unauthorized reproduction or distribution of this program, or any portion of it,
may result in severe civil and criminal penalties, and will be prosecuted to the
maximum extent possible under law.

Welcome to the Brooktrout Setup Program.

Depending on your needs this program will install the Boston API & Driver on your
system.
Press Enter to continue or press ’q’ then Enter to quit setup.
```

6. The program checks for the operating system:

```
The setup program detected Solaris as your operating
system. Is this correct [y or n]?
```

7. Type **y**.

   If the program cannot detect the operating system, it requests that you choose one:

```
If your operating system type is not listed below, then
it is not supported and you may want to exit this
program.
(1) Sunsoft Solaris
(2) SCO UnixWare
(3) Exit Installation
Select your operating system type from the above list:
```

8. Type **1**.

   The program prompts you for the files to install:

```
1) Boston API/Driver Files.
2) Firmware Files.
3) Boston API/Driver Files and Firmware files.
0) Exit Installation.
Please enter the files you wish to install?
```

9. Type the number of your choice.

   The program displays:

```
Default install directory is /usr/sys/brooktrout.

Please enter desired install directory or Enter for
default?
```

10. After you enter an install directory or use the default directory, the program asks:

```
Would you like to configure the driver at this time
[y or n]?
```

11. The program asks:

```
Do you want to continue with the installation of
BOSsetup [y, n?]
```

    If you answer **y**, the program installs the files, configures the driver, and configures the number of channels.

    If you choose Bfv binary and firmware files, the directory ***/usr/sys/Brooktrout*** contains the directories and files shown in *Chapter , Directory Structure on page 147*.

After installing the software, you can download the firmware and configure the call control for the board, see:

- *Chapter , Downloading Firmware on page 245*
- *Using a Call Control Configuration File on page 145*

Removing the Software

To remove all of the files and directories previously installed with ***pkgadd***, use the ***pkgrm*** program.

➢ ***To uninstall both the API and driver, enter:***

```
pkgrm BOSsetup
```

This command removes only the files and directories that the installation procedure placed on the system.

Starting or Stopping the Driver

Because the Boston driver loads automatically when needed, there are no explicit steps required to load the driver.

➢ ***To stop or shut down the driver, enter:***

```
/etc/bostunld -r
```

Reinitializing the Driver

You can reinitialize the driver to its starting state in terms of channels assigned to modules and module number assignments using the ***driver_reinit.exe*** utility located in the ***boston/driver/solaris/user*** directory (see ***Reinitializing the Device Driver on page 285***).

Configuring the Driver Manually

➢ ***To configure the driver manually, change to the installation directory and run the dinstall program:***

The configuration dialog is as follows:

```
Installing Brooktrout Boston Device Driver Version 5.2.0
Maximum number of PCI/cPCI hardware modules (default 16):
Physical buffer size (default 32768):
Application buffer size (default 10240):
Machine ID, in hex (default 1):
Do initial reset (default 1):
History enable (default 0):
```

If you enter ***1***, ***dinstall*** asks the following questions:

```
History size (default 1024000):
Number of physical channel histories (default 0):
Number of application channel histories (default 0):
Configure advanced parameters (n)?
```

If you enter ***y***, ***dinstall*** asks the following questions:

```
Number of I2O frames per module (default 16):
Driver flow control interval (default 250):
Internal memory allocation minimum (default 152):
Internal memory allocation quanta (default 128):
To reserve module numbers, enter mod_id (serial number)
and mod_num, in hex. Mod_num will default as indicated:
mod_id mod_num[F0]:
```

```
Configure test parameters (n)?
```

If you enter ***y***, three test options are listed:

```
Test1:
Test2:
Test3:

Configuring Boston driver:
   Phys buf size 32768
   App buffer size 10240
   Machine ID 1
   History size 1024000
   Num phys hist bufs 0
   Num apl hist bufs 0
```

Leave the maximum number of PCI hardware modules at its default. However, if fewer modules are in use, you can reduce this number to conserve memory. Each hardware module uses approximately 60K.

History Logging

When the Boston driver is installed, history logging is disabled by default. When you manually install the driver using ***dinstall***, the default is set to disable history logging (which you can choose to override). History logging is a powerful debugging tool, especially when working with Technical Support. However, it does consume a significant amount of CPU time when running very high call loads. Therefore, leave history logging off for systems that have high call loads and are CPU-bound. For low-load systems or when debugging, enable history logging.

Leave the number of physical and application channel histories at 0 unless you are directed to change this by Technical Support. If you change the physical or application buffer size, do not change the value to one smaller than the default.

If you specify a number for the physical channel histories, ***dinstall*** asks if you want to restrict the physical channels to administrative channel 1:

```
Restrict physical channels to admin channel 1 (default 1):
```

Choose the default (enable).

Advanced Configuration Parameters

If you selected advanced configuration parameters, the software prompts you for the number of I2O frames, driver flow control interval, internal memory allocation parameters, module reservations, and test parameters. Do not configure advanced parameters except under the advice of Dialogic Technical Support.

The driver is configured and ready to be started.

Resource Allocation Failures

Depending on the options chosen and the state of the system, various resource allocation failures can occur. The errors are displayed and logged, but because of the architecture of Solaris drivers, the installation procedure has no way to know that the driver load failed.

Removing the Manually Configured Driver

To uninstall the driver, navigate to the ***/usr/sys/brooktrout/boston/driver/solaris/install*** directory and enter the following at a command line:

`./dinstall -r.`

Using Bostsrv Under Solaris

When a developer needs to run a multi-process application, you must use the Boston Host Service (Bostsrv). Bostsrv is installed and started automatically by the Configuration Tool before you start your own application. The command line process is not required.

Note: You need to use the Boston Host Service, but it is optional for other Dialogic modules.

When you use the Boston Host Service (Bostsrv) with a Solaris system the following syntax applies:

- Go to following directory:

```
cd /usr/sys/brooktrout/boston/bfv.api/solaris/bin/Bostsrv
```

- Use any of the following command line options:

Option	Meaning
-c <file>	Full path to the btcall.cfg file and name of the btcall.cfg file
-D	Run as daemon (in background)
-f <path>	Full path to firmware
-k	Stops the daemon
-stop	Stops the daemon
-query	Queries whether the server is running
-debug	Console app/debugging
-v	Enable Bfv API debug mode

Note: **-c** and **-f** are required. For example:

```
bostsrv -c /usr/sys/brooktrout/boston/config/btcall.cfg
-f /usr/sys/brooktrout/boston/fw
```

Root login for Unix

When using Unix, you must log in as root to run Bostsrv from the command line in debug mode. Bostsrv creates a process id file in the root `owned/var/run` directory. This prevents multiple copies from running at the same time. It also supports the kill, stop, and query operations.

Special Considerations

When you install the software on a Solaris operating system, consider the following:

PCI Considerations

Dialogic currently only provides PCI boards in the TR1000 Series and TruFax® models. When configuring the device driver, you must select 0 when asked for the number of ISA boards.

The operating system and/or system BIOS assigns base addresses and interrupt numbers to the PCI boards, which might change when the system is rebooted, especially if devices are added or moved.

If multiple PCI boards are in use, to determine the module numbers of boards or ordinal channel values, see ***Determining the Board Module Number on page 2*** for more information about modules and module number assignment.

On some systems, when certain Dialogic® Brooktrout® boards are installed the system might hang at boot time if the device driver is not already installed. Remove the boards, install the driver, then reinstall the boards.

Bfv API Libraries and Makefiles

The Dialogic® Brooktrout® Bfv API libraries are supplied in several forms. 32-bit versions are always supplied, and for Solaris SPARC version 7 and higher, there are also 64-bit versions. The primary libraries are shared (.so) libraries. There are also static (.a) libraries; their use is deprecated and they are supplied for compatibility only.

To compile an application with the ***.so*** libraries, the operating system needs to find the ***.so*** libraries at runtime. The installation procedure automatically installs these libraries into a known system locations: `/usr/lib` (for 32-bit) and `/usr/lib/sparcv9` (for 64-bit). If you are performing installation manually, you can either copy the libraries there, or add the directories containing the ***.so*** libraries to the environment variables LD_LIBRARY_PATH (for 32-bit) and LD_LIBRARY_PATH_64 (for 64-bit).

The ***app.src*** and ***bapp.src*** directories contain makefiles that compile using the 32-bit version of the Bfv API libraries. There is also a sections of the makefile that can be uncommented to use the 64-bit versions of the library. Follow the instructions in the makefile to make these changes.

All applications should try to follow the makefiles supplied in the `app.src` or `bapp.src` directory. In particular, since all the supplied libraries were compiled as multi threaded, compile all applications this way. See also, ***Reviewing Compiler and Linker Options on page 43***.

Writing Multithreaded Programs

When developing a multithreaded program, be sure to follow the programming guidelines described in your system documentation.

When new threads are created, often a stack size can be specified. The stack size must be sufficiently large to store all local variables used both by the application and the API. If the stack size is not large enough, unpredictable behavior can result.

Dialogic usually finds a stack size of 64K to be sufficient but, depending on the application stack allocation demands, more memory might be required.

Warning and Error Messages

If the driver has an error or warning to report (often related to driver startup problems), the message always appears on the system console if such a device exists for Solaris. In addition, message are available in other ways. A limited amount of messages are sometimes available if you use the `crash` command selecting the `panic` function within. The messages are logged to a text file: ***/var/log/messages***.

Memory Allocation Error

On some systems, the driver might fail to load with a memory allocation error appearing on the console. This could happen either as part of running ***dinstall*** or the first time running a Bfv API application since booting the system.

This error occurs because the system's kernel virtual memory space becomes fragmented over time and cannot provide space for the amount of memory required. This error is especially likely if you chose a large history size.

If the driver is loaded soon after boot time (by running any application that uses the driver), the chances of being able to allocate the memory are much higher.

pci_pci Loading Message

On some systems, at boot time a message might appear on the system console mentioning forced loading of pci_pci. This message can safely be ignored.

Determining Installation Parameters

Any time after the driver has been installed (but not removed), you can determine what installation parameters were used by looking at the file ***/platform/<arch>/kernel/drv/boston.conf***. Where ***<arch>*** is ***i86pc*** on x86 platforms and ***sun4u*** on SPARC platforms.

Command Line Configuration for dinstall

When configuring the device driver using ***dinstall***, you can specify the command line option `-c` to cause the ***dinstall*** program to not prompt you for any parameters. Instead, the ***dinstall*** program uses the responses that you supply following the `-c` option on the command line.

If you use the `-c` option, you must include all the parameters described below, in order, regardless of whether or not the parameter has meaning on the system in question. If a parameter does not apply, it is ignored. The parameters are not checked for validity.

Note: Prompting for removal of old driver versions is not performed.

When configuring the driver using ***dinstall***, you must include the following parameters:

Parameter	Defaults
Reserved	0
Reserved	-1
Max num PCI/CPOS hardware modules	16
Physical buffer size	32768
Application buffer size	10240
Reserved	0
Machine ID	1
Reset option (1 or 0)	1
History Enable	0
History Size	1024000
Num physical histories	0
Num app histories	0
Restrict phys channels (1 or 0)	1
Number of I2O frames	*0
Flow control interval	*0
Memory alloc minimum	*0
Memory alloc quanta	*0
Reserved	0
Test parameter 1	*0
Test parameter 2	*0
Test parameter 3	*0

Reserved module info module id, module number 0 to terminate	*0
Reserved	0
Int Prio (Reserved on SPARC)	0

The advanced parameters, marked with * in the list above, can always be safely set to 0. This setting tells the driver to use the default values.

There are currently no parameters used with `-c` when removing the driver, but the option might still be given on the command line for future use.

For example, to configure the driver for 0 ISA hardware modules and up to 16 PCI hardware modules use the following settings:

- physical buffer size 32768
- application buffer size 10240
- machine ID 1
- reset on
- history enabled
- history size 1024000
- physical histories 0
- application histories 0
- restrict physical channels
- all default advanced parameters

Enter the following at the command line:

```
./dinstall -c 0 -1 16 32768 10240 0 1 1 1 1024000 0 0 1 0 0 0 0 0 0 0 0 0 0 0
```

Structure Packing

The API uses packing, via the `#pragma pack` directive, to ensure that certain structures are packed at known correct boundaries. Certain compilers (including some versions of the C++ compiler for Solaris) seem not to handle packing properly. When using compilers with this problem, the best solution is to write source code modules that use Dialogic® Brooktrout® functions and structures in C. Then, if desired, write other portions in C++.

Compiler Issues

This version of the API was compiled with version 4.2 or later of the Sun C compiler (Sun Workshop C/C++ 4.2). Object files and libraries compiled with these versions of the C compiler might be incompatible with those compiled with earlier versions.

To ensure proper functioning of your applications when using an older version of the C compiler, Dialogic recommends that you contact Technical Support for the source version of the API and compile the API library using the older compiler.

After installing your Sun C compiler, be sure that your paths are set up correctly before you begin working with the Dialogic® Brooktrout® software. Failure to do so could cause accidental use of incorrect default compiler components resulting in many errors.

Driver File Locations

When using ***dinstall*** to configure the device driver, you need not have all files from the distribution. However, the locations of files used by the installation procedure relative to the directory from which ***dinstall*** is being run must be the same as that in the normal distribution.

Assuming a directory named ***install*** from which ***dinstall*** is being run, the following files must be present:

install/dinstall
install/dinstlib
kernel/27/boston32
kernel/27/boston64 (SPARC only)
kernel/bostunld

These files and directories appear in the standard distribution under the directory ***driver/solaris***.

Reviewing Compiler and Linker Options

The makefiles contained in the ***app.src*** and ***bapp.src*** directories provide a number of command line options to the compiler and linker. Using these options, the compiler and linker produce object files and executables compatible with the Bfv API library.

Dialogic strongly recommends that you use these options for all your Bfv API applications. But for those who want to change the set of options the compiler and linker use, the following is a list of the nonstandard options and their functions. The options used for compilation are marked with (***C***) and those for linking are marked with (***L***).

`-DSOLARIS`	(***C***) Define that indicates that Solaris is running.
`-D_REENTRANT`	(***C***) For multithreaded programs.
`-misalign`	(***L***) Solaris SPARC only. Allows the program to work with packed structures.
`-lthread`	(***L***) Multithreaded library.
`-xCC`	(***C***) The level of ANSI conformance.
`-xarch=v9 -xregs=no%appl -xO3`	(***C***,***L***) Solaris SPARC 64-bit only. The three of these options together instruct the compiler/linker that 64-bit should be used in a manner consistent with Brooktrout's API libraries.
`-lCstd -lCrun`	(***L***) C++ libraries. Required when new call control is used.

4 - Installing the Developer Kit on Windows®

This chapter describes how one can install Dialogic® Brooktrout® software on Windows® operating systems.

This chapter contains the following sections:

System Requirements for Installation

When running the installation, you need the following:

- Administrator privileges.
- Sufficient available disk space on the drive that contains the Windows® temp directory to extract the ***sdk_windows.exe*** file.

  By default the Windows® operating system sets it to [WINDOWS_VOLUME]\temp, where WINDOWS_VOLUME is the drive where the operating system is installed. If you do not have enough space, you will need to adjust the TEMP environment variable to point to a drive that contains at least 200 MB of free space.
- An Intel x86-based computer that contains an Intel x86 (including Pentium) CPU only.
- The minimum system specification for running the TR1000/TR1034/SR140/TruFax® Developer Kit SDK is based on the number of channels running on the system. For software only solutions, the maximum CPU utilization the Dialogic® Brooktrout® HBF software can use when running a number of fax channels simultaneously is shown in ***Table 1***. For hardware only solutions, the minimum processor and memory requirements are shown in ***Table 2*** for Intel based systems, and ***Table 3*** for SPARC based systems.

Table 1. Maximum CPU Utilization

Processor	4 Lines	8 Lines	24 Lines	48 Lines	120 Lines
Pentium 4 - 2.66 GHz	3%	5%	16%	33%	80%
Pentium 4 - 3.4 GHz	2%	4%	12%	20%	60%
Dual Pentium 4 - 2.66 GHz	2%	3%	10%	20%	45%

Table 2. Intel System Performance Requirements

Maximum No. of Channels	Processor	Memory	Maximum No. of Boards
120	Celeron 1 GHz.	128 MB.	1
192	Pentium 4 2.26 GHz.	256 MB.	2
360	Pentium 4 2.4 GHz.	512 MB.	3
480	One or Two Xeon 2.0 GHz	2 GB.	4

Table 3. SPARC System Performance Requirements

Maximum No. of Channels	Processor	Memory	Maximum No. of Boards
120	UltraSPARC IIIi 1.06 GHz.	512 MB.	1
192	UltraSPARC IIIi 1.28 GHz.	512 MB.	2
360	UltraSPARC IIIi 1.28 GHz.	1 GB.	3
480	One or Two UltraSPARC IIIi 1.28 GHz	2 GB.	4

Installation Packages

There are two installation packages for Windows. This chapter describes the SDK installation (***sdk_windows.exe***), which is for application developers.

The installation package for end users (***boston.msi***) is described in the ***Developer Guide, Packaging Your Application for Windows***. This chapter explains how you can integrate and redistribute the runtime software, also called Dialogic® Brooktrout® System Software, with your own installation procedures for your users.

Installing the SDK

Ordinarily, you install the SDK from the distribution CD. However, there are other ways that the installation can be launched. For example, you can also install it from the Dialogic web site.

Also, although Dialogic recommends that you install the SDK before you install your Computer Telephony, if you choose to install the boards first, the Found New Hardware Wizard searches for and launches the installation. If the Wizard cannot find ***sdk_windows.exe***, it prompts you to browse and locate the installation software.

If you have Host Based Fax (SR140) modules, you must install the TR1000/TR1034/SR140/TruFax® Developer Kit SDK software first. Then you can turn on SR140 modules using the Dialogic® Brooktrout® License Manager. See the ***Dialogic® Brooktrout® SR140 User Guide***.

Note: If you have a previous version of the SDK installed on your system, you must uninstall it before you can successfully install SDK 5.2.0. The InstallShield Wizard does not install SDK 5.2.0 if there is a previous version on your system; however, it notifies you of the previous version and suggests corrective action. See ***Removing or Modifying the Software on page 54*** for more information.

➢ ***To install the TR1000/TR1034/SR140/TruFax® Developer Kit from the CD-ROM:***

1. Insert the Developer Kit CD-ROM into a CD drive.

   The Autorun program starts automatically.

   If the Autorun program does not start automatically:

   a. Go to the CD root directory on the CD-ROM.

   b. Launch ***autorun.exe***.

Autorun Menu

The Autorun Menu appears:

2. Choose Install SDK from the list of options, which starts the setup procedure and runs ***InstallShield*** to extract the files needed to install the SDK on your system.

   The splash screen followed by the Welcome window appears.

3. Click Next.

   The License Agreement window appears.

4. Read the license agreement and, if you agree, select the appropriate option and click Next.

   The Destination Folder dialog box appears.

5. Click Change to choose a folder or click Next for the default folder, ***C:\Brooktrout\Boston***.

   In the example, ***C:\Brooktrout\Boston*** is the destination folder where the application software installs the SDK contents.

   If you choose Next, go to ***Step 7.***

   If you choose Change, the Change Current Destination Folder dialog box appears.

6. Click OK after selecting a new location for the software, which takes you back to the Destination Folder dialog box.

7. Click Next on the Destination Folder dialog box.

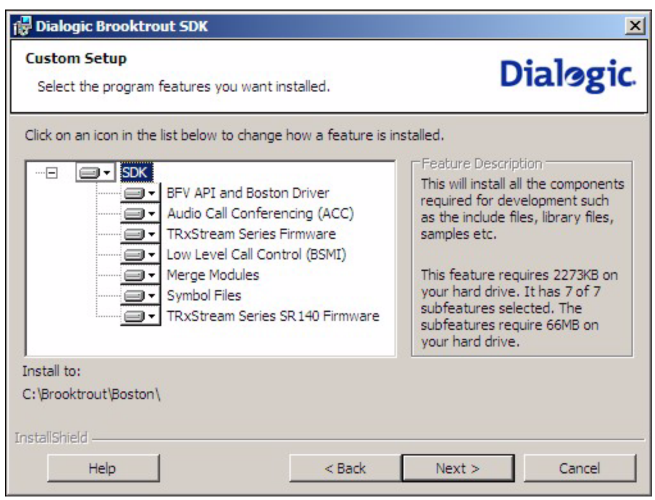
   The Setup Type dialog box appears.

8. Choose one of the setup types:

   - Select Complete for the full software development kit that includes the Bfv and other APIs used for writing your applications, the tool to configure your boards and software, the runtime system, and troubleshooting tools.
   - Select Custom for picking specific software elements that exactly match your development needs.

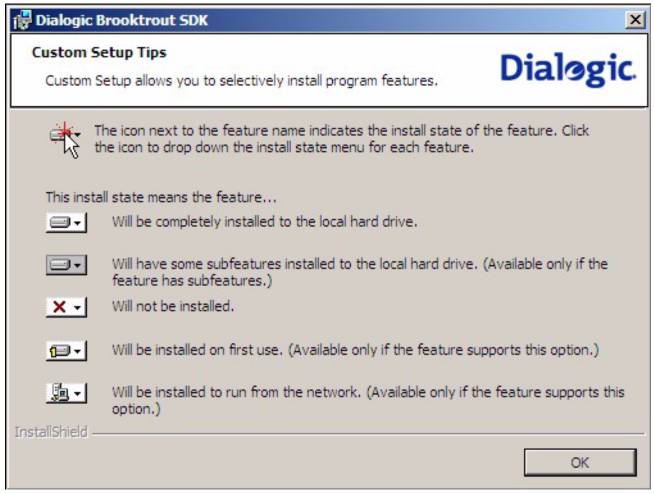
Custom Setup

If you choose Custom, the Custom Setup window appears.



Custom Setup Tips

For help, click Help to see the Custom Setup Tips window.



9. After reading the Help, click OK to return to the Custom Setup window.
10. Select the components you want to install and click Next.

    If you choose Complete or if you select a custom installation, the Ready to Install window appears.
11. Click Install to install the software.

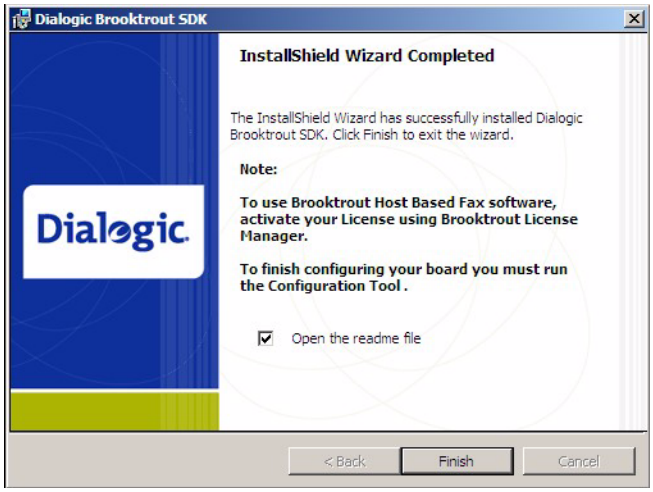
    The Installing Brooktrout Boston SDK window appears.

    After the software is installed, the InstallShield Completed windows appears.

    The installation program reminds you to install your Computer Telephony using the Add New Hardware Wizard.
12. Click Finish.

InstallShield Wizard Completed

When the Windows® installation finishes, the following InstallShield Wizard Completed window appears:



13. Shut off your computer and install the board using the instructions in the hardware installation guide that came with the board.

Installation Directory

When the installation is complete, the resulting directory, ***[INSTALLDIR]:*** (***C:\Brooktrout\Boston*** is the default), contains the directories according to the options you choose. See ***Chapter , Directory Structure on page 147*** for the installed directories and files.

Installing Licenses (SR140 Only)

14. After rebooting your system, you can install your Dialogic® Brooktrout® license. See the ***SR140 User Guide*** for more information.

Sample Applications

15. You must compile the sample applications if you want to use them. See the chapter in the ***Developer Guide*** on compiling the sample applications.

Interrupted Installation

The InstallShield Wizard does not install SDK 5.2.0 if you have a previous version. However, it notifies you of the previous version and suggests corrective action.

You can also cause this interrupted window to appear for a variety of reasons including:

- When an installation in progress is halted
- System crash or other hardware malfunction

See ***Removing or Modifying the Software on page 54*** and ***Removing the Plug-and-Play Driver on page 68***.

Removing or Modifying the Software

After you install SDK 5.2.0, you can modify, repair, or remove it by running ***sdk_windows.exe*** again. Modifying or removing SDK 4.1 means reinstalling the software to add or delete functionality.

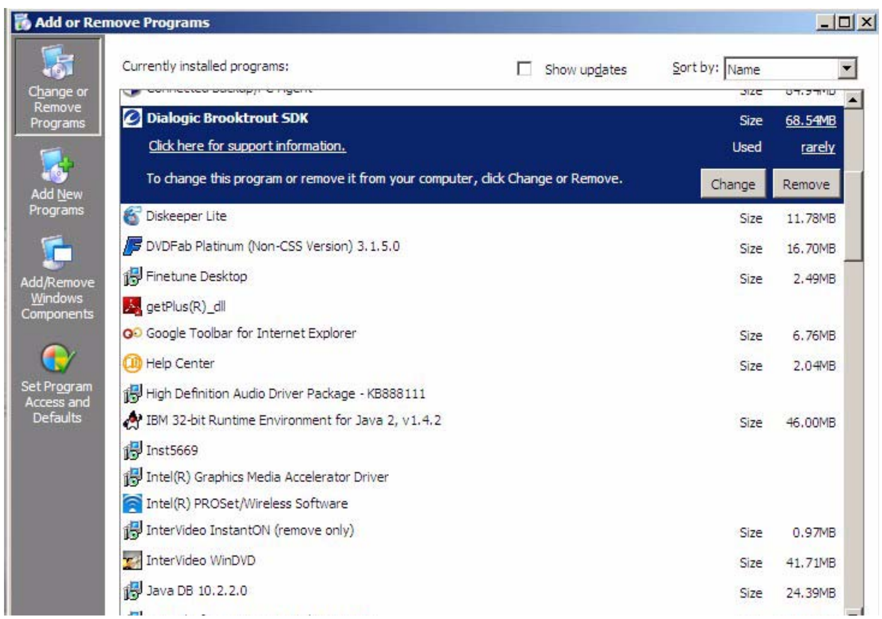
You can also remove or change SDK 5.2.0 or a previous version of the SDK by using Add/Remove Programs from the Control Panel.

➢ ***To uninstall or modify the existing software from the Control Panel:***

1. Click either the following:
   - My Computer → Control Panel → Add/Remove Programs
   - Start → Settings → Control Panel → Add/Remove Programs

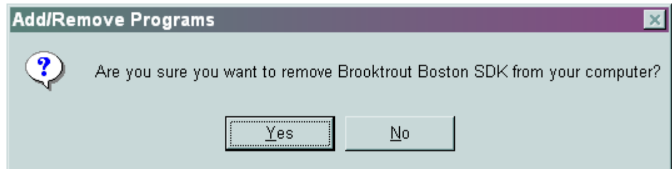
Add/Remove Program

The Add/Remove Programs window appears.



2. Select Brooktrout Boston SDK from the program list.

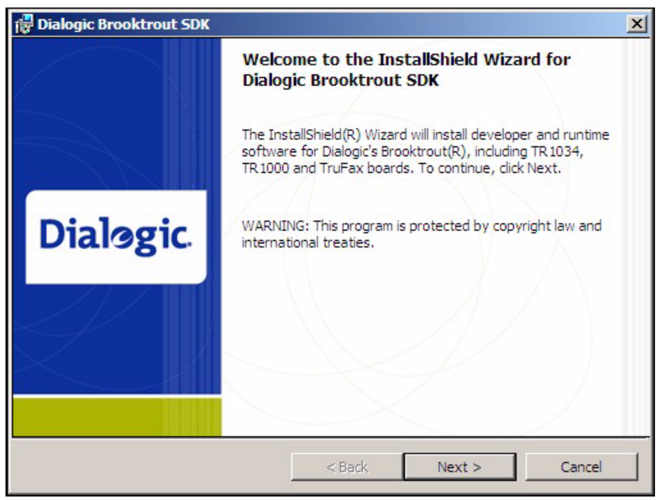
   If you click Remove, the system asks you if you want to remove the software in the Add/Remove Programs dialog box.



3. Click Yes.
4. If you click Change in the Add/Remove Program window, the Welcome window from the software installation appears.

Dialogic® Brooktrout® Welcome

This is the same window that you get if you run ***sdk_windows.exe*** again. For details, see ***Installing the SDK on page 48***.



5. Click Next.

   The Program Maintenance window appears.
6. Choose one of the options and click Next.

   If you choose Modify, the Custom Setup window appears.
7. Select the components you want to modify and click Next.

   The Ready to Modify the Program window appears.

8. Click Install.

   The software installs the Dialogic® Brooktrout® SDK.

   If you choose Repair, the Ready to Repair window appears.

9. Click Install to have the wizard install the Dialogic® Brooktrout® software again.

   Then the InstallShield Wizard Completed window appears after the software is reinstalled to either modify or repair it.

   If you choose Remove, the Remove the Program window appears.

10. Click Remove.

    The Uninstalling Brooktrout Boston SDK window appears.

    When the removal of the software is complete, the InstallShield Wizard Complete window appears.

11. Click Finish to complete the operation.

These procedures only remove the files and directories that the installation procedure placed on the system.

5 - Installing the Device Driver on Windows®

This chapter describes how one can install the device driver on Windows® systems.

This chapter contains the following:

Installing and Configuring Options

Once the computer telephony hardware and software are installed on your system, you must install and configure the device driver for the Dialogic® Brooktrout® boards. This chapter describes two methods for installing the device driver:

- Automatically by letting the operating system identify the device, find the driver, and install it. See ***Installing the Plug-and-Play Driver on page 59***.
    - You must be running on Windows® 2000 (or later) to install the driver automatically using the Plug-and-Play (PnP) feature of the operating system.
    - SR140 (virtual module) users can also use the Dialogic® Brooktrout® Configuration Tool to install the driver.
- Manually running the ***install*** program from the command line. See ***Installing and Configuring the Driver using a Command (cmd) Prompt on page 72***.

  You can install the PnP driver manually using the ***install*** program with Windows® 2000 (or later).

Note: For Windows Vista® users: If you do not run the driver installation program with administrative privileges, a dialog box will appears requiring you to respond to continue the installation.

Installing the Plug-and-Play Driver

After you have installed your Dialogic® Brooktrout® board the operating system can automatically install the driver for your board.

The Microsoft hotfix 909666 is required when running Windows® 2003 x64. Obtain the hotfix from Microsoft at http://support.microsoft.com/ by referencing the knowledge base number 909666. System crashes can occur without this hotfix.

Note: The dialog boxes for Windows® 2000 Plug-and-Play driver installation are somewhat different from those for newer versions of Windows® systems. The dialog boxes shown in this chapter show Plug-and-Play driver installation on a Windows 2000 system.

➢ ***Do the following to install the driver for your board:***

1. Turn on your computer.

   As the operating system finishes booting, the Found New Hardware Wizard window might appear.

Found New Hardware Wizard

This window does not appear if the hardware was already detected from a prior boot or if you are upgrading from an older installation.

See ***Updating Computer Telephony and the Plug-and-Play Driver on page 66*** for information about upgrading.

2. Click Next to continue the device driver installation.

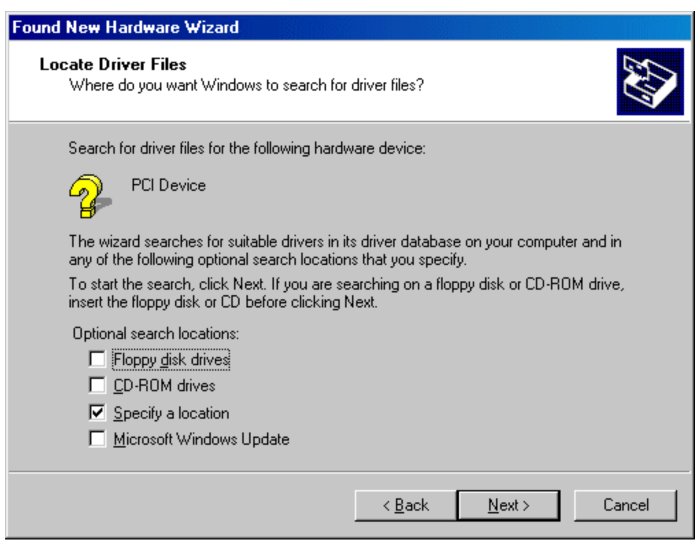
Installing Hardware Device Driver

The Install Hardware Device Driver window appears.

3. Choose the Search for a suitable driver for my device (recommended) and click Next.

Locating Driver Files

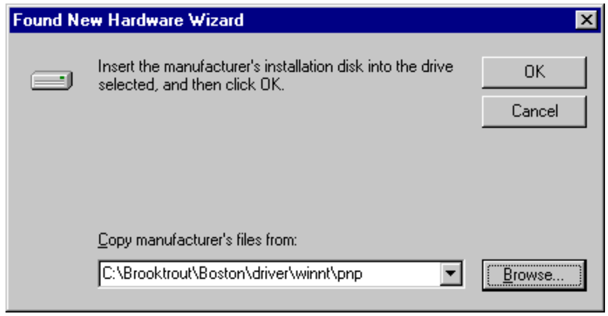
The Locate Driver Files window appears so the wizard can search for a suitable driver.



4. Check Specify a Location and click Next.

Browsing for the *.inf* File

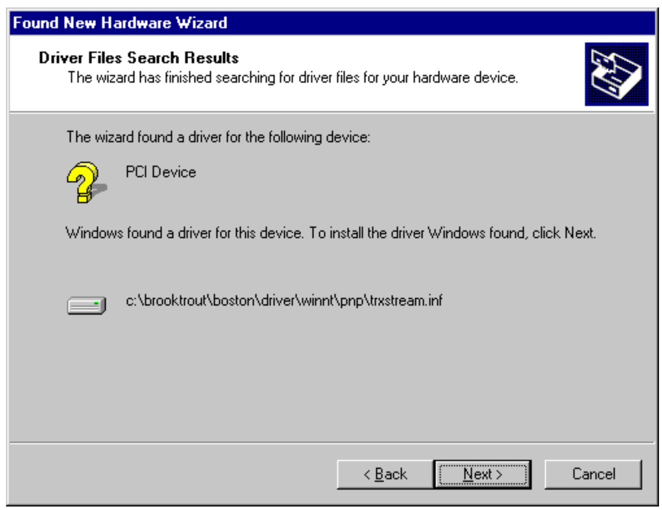
5. The following window appears where you can browse to select the ***trxstream.inf*** file located in the following folder: C:***Brooktrout\Boston\driver\winnt\pnp***



6. Click OK.

Finding the Driver File

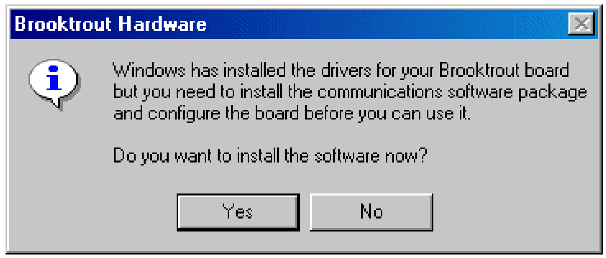
The Driver File Search Results window appears.



7. The Hardware Co-Installer checks if the Dialogic® Brooktrout® Software is installed on your system.

   a. Click Next to install the Dialogic® Brooktrout® Software.

   If you do not have Dialogic® Brooktrout® Software installed, before the Completing the Found New Hardware Wizard window appears, the Computer Telephony dialog box provides these options:



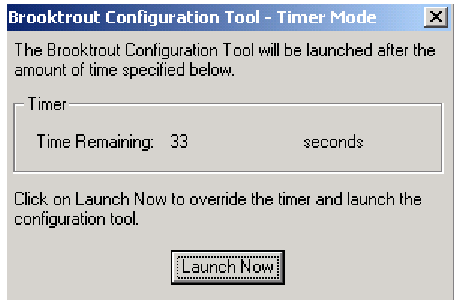
- Click Yes to install the Dialogic® Brooktrout® Software.
- Click No to install the Dialogic® Brooktrout® Software later.

  Follow the directions on the Install Path dialog box, then click OK to install the software.

Note: Restart the system when prompted by the final step in the SDK installation.

   b. The Hardware Co-Installer window and the Dialogic® Brooktrout® Configuration Tool Timer dialog boxes appear.

c. Click the Launch Now button in the Timer Mode dialog box to set a timer that launches the Configuration Tool after a specified amount of time.



Do not click the Launch Now button if you have more than one Dialogic® Brooktrout® board. Instead, finish the Found New Hardware Wizard installation. The Windows® system continues to detect and install the remaining Dialogic® Brooktrout® boards. The timer automatically resets for each successive installation.

Refer to the ***Plug-and-Play Installation Scenarios*** in the ***Developer Guide***.

8. Click Next to the Co-Installer window.

   The Completing the Found New Hardware Wizard window appears.

9. Click Finish.

   The Plug-and-Play installation is complete.

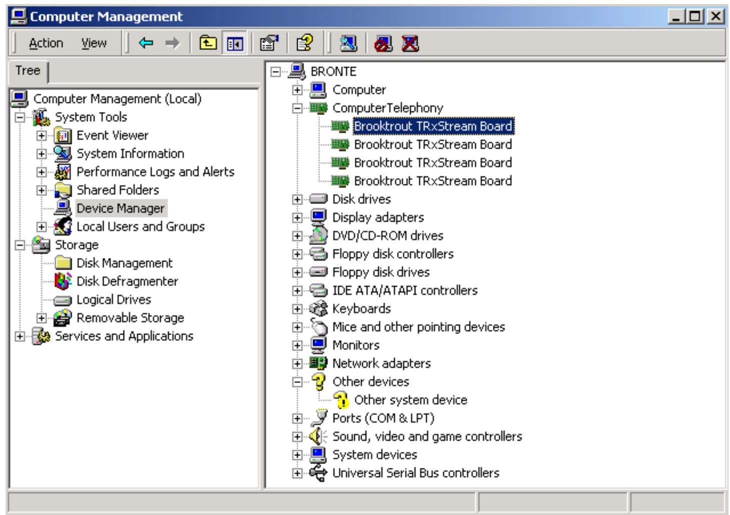
Verifying Dialogic® Brooktrout® Board Integration

➢ ***To verify that the Dialogic® Brooktrout® board is successfully installed on your system:***

1. Click one of either:

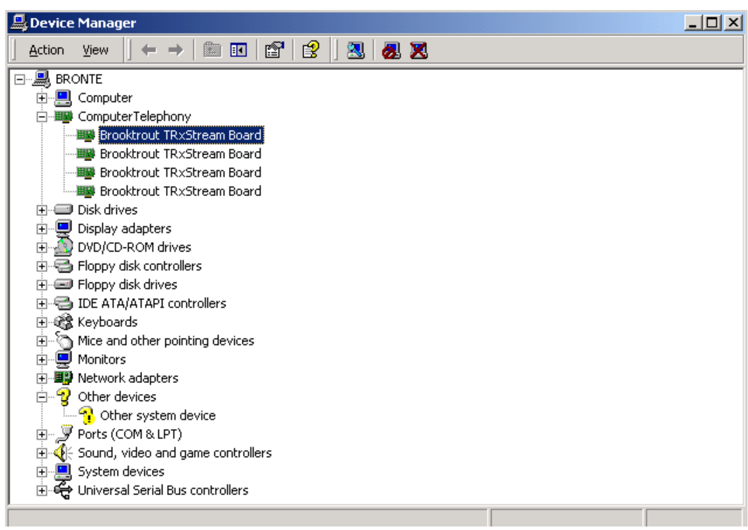
   - Control Panel → Administrative Tools → Computer Management

     The Computer Management window appears:



   - Control Panel → System → Hardware → Device Manager

     The Device Manager window appears:

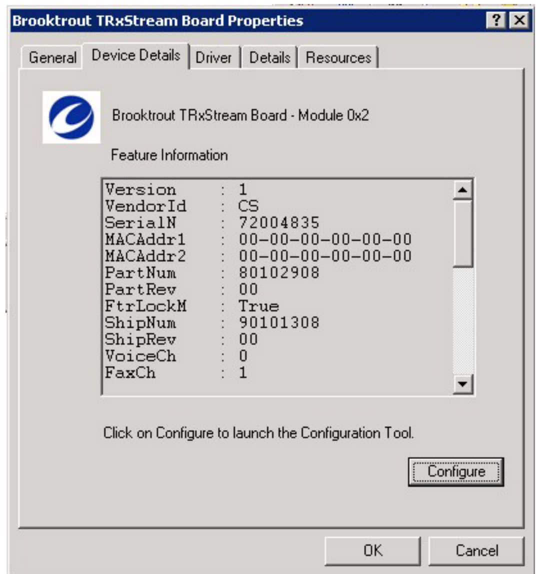


2. To view the properties of the Dialogic® Brooktrout® board in your system, double-click the board name in one of the windows shown above.

Viewing Dialogic® Brooktrout® Board Properties

The Brooktrout TRxStream Board Properties dialog box appears.

3. Click the Device Details tab.



4. Click the Configure button to use the Dialogic® Brooktrout® Configuration Tool Wizard to configure the device driver as well as other facilities of the board (for example, telephony).

   For extra help, see *Chapter , Configuring Software on page 89* for information about configuring your system. The Configuration Tool starts in Wizard mode if the boards are identical, or Advanced Mode if the boards are different.

Updating Computer Telephony and the Plug-and-Play Driver

A Plug-and-Play-compliant device driver cannot be started unless the devices managed by the driver are physically installed in the host system ***and*** are in an enabled state. SDKs prior to SDK 3.1 did not ship with a Plug-and-Play-compliant device driver.

Removing Dialogic® Brooktrout® Hardware

➢ ***To remove the previous version of the Computer Telephony and device driver:***

1. Click Start → Settings → Control Panel.

   Select the Add/Remove Hardware option to remove the previously installed Dialogic® Brooktrout® boards.

   The Add/Remove Hardware Wizard window appears.
2. Click Next.

   The Choose a Hardware Task window appears.
3. Select Uninstall/Unplug a device and click Next.

   The Choose a Removal Task window appears.
4. Select Uninstall a device and click Next.

   The Installed Devices window appears.
5. Select the device labeled PCI Device (or possibly Network Controller for some older digital boards) and click Next.

Note: If you have more than one device named PCI Device listed under Devices, select the first one.

   The Uninstall a Device window appears.

6. Select Yes and click Next.

   The Add/Remove Hardware Complete window appears and informs you that the device was successfully removed.
7. Click Finish.
8. Follow Step 1 through Step 6 to remove each Dialogic® Brooktrout® board that is listed as PCI Device (or possibly Network Controller) in the Devices list in the Installed Devices window.

Once the Plug-and-Play installation process is completed, verify that the Computer Telephony category appears correctly in the Device Manager and contains all installed Dialogic® Brooktrout® hardware.

See ***Verifying Dialogic® Brooktrout® Board Integration on page 64***.

Updating Previous Versions of the Plug-and-Play Driver

After you update the driver, reboot your system to ensure that the new driver is used.

➢ ***To update the device driver, follow these steps:***

1. Open the Windows Device Manager.
2. Select Start → Settings → Control Panel → Administrative Tools → Computer Management.
3. Select the Device Manager node.
4. Expand the Computer Telephony node.
5. Right-click a Dialogic® Brooktrout® board and select Properties to open the board properties window.
6. Select the Driver tab and look for the Update Driver button.
7. Click Update Driver to start the Update Driver Wizard.
8. Click Next.
9. After the Install Hardware Device Drivers window appears, select Search for a suitable driver for my device.
10. The Locate Driver Files window appears, select Specify a location.
11. On the next window, choose Browse and locate the ***trxstream.inf*** file located on the CD under the ***Windows Plug-and-Play*** folder, or if you have already installed the SDK, locate the file under ***[INSTALLDIR]\driver\winnt\pnp***.

    If the specified location for the new ***trxstream.inf*** file is different from the location for the old ***trxsteam.inf*** file, and if the file still exists in the old location on your system, the update does not work. The system uses the old file and location instead of the new one. Delete the old copy of the ***trxstream.inf*** file and its Plug-and-Play components before installing the new one.

    If you do not have the software installed, the Co-Installer provides an option to install the software. It also launches the Configuration Tool in Timer mode. You can wait the specified amount of time before launching the Configuration Tool, or launch it immediately in Advanced Mode.
12. Click OK and then Next.
13. Click Next and then Finish.

Removing the Plug-and-Play Driver

Installing a Dialogic® Brooktrout® device using the Plug-and-Play Manager (Found New Hardware Wizard) creates some backup files and registry entries. The Windows® Plug-and-Play Manager uses this information to automatically install the device on rebooting.

To completely remove the Plug-and-Play driver from the system, you must perform a complete cleanup after you remove the device from the Device Manager.

Make sure you stop all your applications and the Boston Host Service before uninstalling the device driver.

Note: The following instructions include steps to take when removing the plug and play driver for versions earlier than 5.2 and for removing the plug and play driver for the 5.2 version released with SDK 5.0.0.

For Earlier Versions of the Driver (Prior to 5.2)

➢ ***To remove an earlier version (prior to SDK 5.2.0) of the Plug-and-Play driver, do the following:***

1. Open Windows Device Manager.
2. Expand the Brooktrout Hardware node.
3. Remove all the Brooktrout boards listed under the Brooktrout Hardware node.
   a. Expand the “Computer Telephony” node.
   b. Right click the Brooktrout board node and select.
   c. Uninstall, Confirm Device Removal, and click OK.
4. Open Command Prompt and type “net stop boston” to stop the Boston driver that can still be running.
5. Delete the following files:

   `C:\WINNT\SYSTEM32\DRIVERS\boston.sys`

   `C:\WINNT\SYSTEM32\brktBdevpp.dll`

   `C:\WINNT\SYSTEM32\brktBdevco.dll`

6. Examine the registry value `InfPath` located under:

   `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{50CE2010-E61B-40EF-9EAA-2BCDE74F8C6C}\0000.`

   This value contains the name of the backup copy of the ***trxstream.inf*** file, that Windows® created during device installation under the ***C:\WINNT\INF*** directory.

Removing the Plug and Play Driver

7. Delete the ***INF*** file of this name from ***C:\WINNT\INF*** along with the corresponding ***PNF*** file. Except for the extension, the ***INF*** file and the ***PNF*** file have the same names. For example, if the ***INF*** file is ***oem11.inf***, the name of the ***PNF*** file is ***oem11.pnf***.
8. Delete the following registry keys:

   `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{50CE2010-E61B-40EF-9EAA-2BCDE74F8C6C}`

   `HKEY_LOCAL_MACHINE\SOFTWARE\Brooktrout Technology\DeviceCoInstaller`

   `HKEY_LOCAL_MACHINE\SOFTWARE\Brooktrout Technology\DevicePropertyPage`

   `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\boston`
9. Reboot the system and verify that the Brooktrout Hardware node is not listed in the Device Manager.

For Driver Version 5.2 (SDK Version 5.2.0.)

➢ ***To remove the Plug-and-Play driver, do the following:***

1. Open Windows Device Manager.
2. Expand the Computer Telephony node.
3. Remove all the Brooktrout boards listed under the Computer Telephony node.
   a. Expand the “Computer Telephony” node.
   b. Right click the Brooktrout board node and select.
4. Open Command Prompt and type “net stop boston” to stop the Boston driver that can still be running.
5. Delete the following files:

   `C:\WINNT\SYSTEM32\DRIVERS\boston.sys`

   `C:\WINNT\SYSTEM32\brktBdevpp.dll`

   `C:\WINNT\SYSTEM32\brktBdevco.dll`

6. Examine the registry value `InfPath` located under:

   `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{8CF4CA66-A2CC-48FA-BC1D-6A64E47F6D27}.`

   Search for the first key which contains the following values:

   - DriverDesc=Brooktrout TRxStream Board

   This key should contain the name of the backup copy of the ***trxstream.inf*** file that Windows® created during the device installation under ***C:\winnt\INF.***

7. Delete the ***INF*** file of this name from ***C:\WINNT\INF*** along with the corresponding ***PNF*** file. Except for the extension, the ***INF*** file and the ***PNF*** file have the same names. For example, if the ***INF*** file is ***oem11.inf***, the name of the ***PNF*** file is ***oem11.pnf***.

8. Delete the following registry keys:

   For 32-bit operating system:

   `HKEY_LOCAL_MACHINE\SOFTWARE\Brooktrout Technology\DeviceCoInstaller`

   `HKEY_LOCAL_MACHINE\SOFTWARE\Brooktrout Technology\DevicePropertyPage`

   `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\boston`

   For 64-bit operating system:

   `HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Brooktrout Technology \DeviceCoInstaller`

   `HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Brooktrout Technology \DevicePropertyPage`

   `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\boston`

9. Reboot the system and verify that the Brooktrout TRxStream node(s) is not listed under the Computer Telephony class node in the Device Manager.

Disabling Plug-and-Play Driver Installation

After you install a Dialogic® Brooktrout® board, the Windows® operating system might display a window stating that a new PCI device was found and offer to detect/install a driver for it. Proceed with the automatic detection. The operating system will not find a matching driver. Do not attempt to supply a driver when asked.

➢ ***The detection of new hardware occurs after each reboot unless you perform the following procedure:***

1. When Welcome to Found New Hardware Wizard appears, click Next.
2. Select Search for a suitable driver for my device (recommended).
3. Click Next, uncheck all choices, then click Next again.
4. Select Disable device
5. Click Finish.

➢ ***Another approach is to:***

1. Click Cancel for ALL new devices that are found.
2. Right-click My Computer.
3. Right-click Properties.
4. Right-click Hardware
5. Right-click Device Manager.
6. Look for:? Other Devices.
7. Right-click all devices with:?.
8. Select Disable.

Reinitializing the Driver

You can reinitialize the driver to its starting state in terms of channels assigned to modules and module number assignments using the ***driver_reinit.exe*** utility located in the ***boston\driver\winnt\user*** directory (see ***Reinitializing the Device Driver on page 285***).

Installing and Configuring the Driver using a Command (cmd) Prompt

To install and configure the driver manually, change to the ***[InstallDir]\driver\winnt\install*** directory and run the ***install*** program. You can use this application to install Plug-and-Play drivers.

The configuration dialog is as follows:

```
Installing Brooktrout Boston Device Driver Version 5.2.0
Maximum number of PCI/cPCI hardware modules (default 16):
Physical buffer size (default 32768):
Application buffer size (default 10240):
Machine ID, in hex (default 1):
Do initial reset (default 1):
History enable (default 0):
```

If you enter ***1***, ***install*** asks the following questions:

```
History size (default 1024000):
Number of physical channel histories (default 0):
Number of application channel histories (default 0):
Configure advanced parameters (n)?
```

If you enter ***y***, ***install*** asks the following questions:

```
Number of I2O frames per module (default 16):
Driver flow control interval (default 250):
Internal memory allocation minimum (default 152):
Internal memory allocation quanta (default 128):
To reserve module numbers, enter mod_id (serial number)
and mod_num, in hex. Mod_num will default as indicated:
mod_id mod_num[f0]:
Configure test parameters (n)?
```

If you enter ***y***, three test options are listed:

```
Test1:
Test2:
Test3:
```

```
Configuring Boston driver:
Phys buf size 32768
App buffer size 10240
Machine ID 1
History size 1024000
Num phys hist bufs 0
Num apl hist bufs 0
Load driver with no devices present? (y or n, default n):
When using Boston PCI hardware modules, a system memory
adjustment may be required in order to properly access the
modules. For a small number of modules this is often not
required, but many factors, including other devices that
are or will be in the system, affect this. After this
adjustment is made, the system must be rebooted.
Do you wish to perform this adjustment? (default n):
```

After installing the Plug-and-Play driver, the following message appears:

```
Installing Plug and Play driver.
The Plug and Play driver was successfully installed.
```

The maximum number of PCI hardware modules can be left at its default. However, if fewer modules are in use, this value can be set to a smaller number to conserve memory. Each hardware module uses approximately 60K.

For more information, see:

- ***Command Line Configuration for Install on page 84***
- ***Windows® Operating System on page 278***

History Logging

When the ***install*** program installs the Boston driver, history logging is disabled by default (0). When you manually install the driver using ***install***, the default is set to disable history logging (which you can choose to override). History logging is a powerful debugging tool, especially when working with Technical Support. However, it does consume a significant amount of CPU time when running very high call loads. Therefore, leave history logging off for systems that have high call loads and are CPU bound. For low-load systems or when debugging, enable history logging (1).

Leave the number of physical and application channel histories at 0 unless you are directed to change this by Technical Support. If you change the physical or application buffer size, do not change the value to one smaller than the default.

If you specify a number for the physical channel histories, ***install*** asks if you want to restrict the physical channels to administrative channel 1.

```
Restrict physical channels to admin channel 1 (default 1):
```

Choose the default (enable).

Advanced Configuration Parameters

If you select advanced configuration parameters, the software prompts you for the number of I2O frames, driver flow control interval, internal memory allocation parameters, module reservations, and test parameters. Do not configure advanced parameters except under the advice of Dialogic Technical Support.

Memory Resources

When Dialogic® Brooktrout® PCI hardware modules are installed in the system, a certain type of memory resource might be exhausted or dangerously low if you do not perform a memory adjustment. For small numbers of modules, the adjustment is often not required, but there is no way to predict the point at which it becomes necessary, since each system is different. Need for the adjustment depends on the system configuration, whether other drivers or devices are using this type of memory or will in the future, and possibly other factors. If you choose to perform the system memory adjustment, ***install*** prompts you for the system memory size.

```
Enter the number of MB of memory in the system[32]?
```

If a memory adjustment has been performed by another program, ***install*** prompts you as to how to proceed.

```
Do you wish to   (1) skip adjusting,
                 (2) replace adjustment, or
                 (3) add to the adjustment
```

Proceeding with any of the adjustments in this case might not properly succeed or be reversible later, but the ***install*** program does the best it can to make the adjustment safely.

The program then advises you to reboot.

```
You must reboot the system for the memory
configuration changes to take effect.
```

```
A reboot is recommended. Do you wish to reboot now? [n]
```

The ***install*** application also prompts you to reboot the system if Windows® requires a system restart to complete the Plug-and-Play installation.

Removing the Driver with a Command (cmd) Prompt

To remove the driver, enter the following at the Command Prompt:

`install -r`

You must include the following parameter (see ***Command Line Configuration for Install on page 84***):

```
Reboot (if needed) after memory readjustment (1 or 0)
```

This option successfully removes all the Dialogic® Brooktrout® PnP devices that are installed using the ***trxstream.inf*** file that came with your SDK. If you installed the Dialogic® Brooktrout® PnP hardware using some other INF file that registers the Dialogic® Brooktrout® boards under a class other than Computer Telephony, the ***-r*** option can fail to completely uninstall this device.

Reinitializing the Driver

You can reinitialize the driver to its starting state in terms of channels assigned to modules and module number assignments using the ***driver_reinit.exe*** utility located in the ***boston\driver\winnt\user*** directory (see ***Reinitializing the Device Driver on page 285***).

Using the Boston Host Service

Boston Host Service (Bostsrv) is used to initialize modules. Upon startup it detects Dialogic® Brooktrout® modules, loads FW and initializes the call control. You must use Boston Host Service to run a multi-process application or to use Dialogic® Brooktrout® SR140 Host Based Fax (HBF) modules. Boston Host Service is optional for other Dialogic® Brooktrout® modules.

Bostsrv is installed and started automatically by the Configuration Tool when you click the Apply button. The command line process is not required.

To run the Bostsrv with the Boston driver, use any of these commands.

What you can do with Bostsrv	Command Line
To start the service	**`net start bostservice`**
To stop the service	**`net stop bostservice`**
To install the service	**`bostsrv -install`**
To remove the service	**`bostsrv -remove`**
To run as a console application for debugging	**`bostsrv -debug`**

Note: If you are running Windows Vista®, you must have administrative privileges to run, start, or stop Boston Host Services.

To use the Boston Host Service on:

- Red Hat Linux, see ***Using Bostsrv Under Linux on page 16***.
- Solaris operating system, see ***Using Bostsrv Under Solaris on page 35***.

Installing Bostsrv

➢ ***To install the Boston Host Service, in a Command Prompt window:***

1. Enter the following at the a command prompt:

   `bostsrv -install`

2. Enter name and location of the btcall configuration file. (default shown)

   `C:\Brooktrout\Boston\bfv.api\winnt\bin\btcall.cfg`

3. Enter the location of the firmware files. (default shown)

   `C:\Brooktrout\Boston\fw`

4. Enter the following at the a command prompt:

   `net start bostservice`

   ```
   The Brooktrout Technology Boston Host Service is
   starting...
   ```

Starting Bostsrv

➢ ***To start the Boston Host Service, in a Command Prompt window:***

Enter the following at the a command prompt:

`net start bostservice`

```
The Brooktrout Technology Boston Host Service is
starting...
```

To set the Boston Host Service to start automatically, set startup mode to Automatic using Windows® Services window. This initializes the Dialogic® Brooktrout® modules upon machine starter reboot.

Troubleshooting Bostsrv

System Error 59

When the Boston Host Service fails to start as a service with a system error 59, verify that you correctly entered the configuration file name and path location, as shown in ***Installing Bostsrv on page 78*** for the ***btcall.cfg*** file.

If you do not specify the full path to the protocol for the ***callctrl.cfg*** file, within the ***btcall.cfg file,*** a system error 59 can also occur.

Special Considerations

When you install the device driver on a Windows® operating system, consider the following topics:

- ***PCI Considerations on page 80***
- ***Bfv API Libraries and Makefiles on page 81***
- ***Writing Multithreaded Programs on page 81***
- ***Using BOSTDLLD with Debugging Enabled on page 81***
- ***Determining Installation Parameters on page 82***
- ***Driver Startup on page 83***
- ***Global Variables on page 83***
- ***Limits on Open Files on page 83***
- ***Process Priority on page 84***
- ***Command Line Configuration for Install on page 84***
- ***Command Line Configuration for Install on page 84***
- ***Version Information on page 87***

PCI Considerations

Dialogic currently only provides PCI boards in the TR1000 Series and TruFax® models.

The operating system and/or system BIOS assigns base addresses and interrupt numbers to the PCI boards, that might change when the system is rebooted, especially if devices are added or moved.

If multiple PCI boards are in use, see ***Determining the Board Module Number on page 2*** to determine the module numbers of boards or ordinal channel values.

Bfv API Libraries and Makefiles

Dialogic only supplies dynamically linked C runtime libraries (BOSTDLLD).

The ***app.src*** and ***bapp.src*** directories contain makefiles that compile using DLLD version of the Bfv API libraries. There is also a section of the makefile that can be uncommented to use the DLLS version of the library. Follow the instructions in the makefile to make these changes.

All applications should try to follow the makefiles supplied in the `app.src` or `bapp.src` directory. In particular, since all the supplied libraries were compiled as multi threaded, compile all your applications this way. See also, ***Reviewing Compiler and Linker Options on page 43***.

When using a dynamically linked version of the library, either add the directory name containing the DLL to your PATH environment variable, or copy the DLL into a directory already in PATH. For example:

```
\Brooktrout\boston\bfv.api\winnt\lib
```

Writing Multithreaded Programs

When new threads are created, often a stack size can or must be specified. The stack size must be sufficiently large to store all local variables used both by the application and the Bfv API. If the stack size is not large enough, unpredictable behavior can result.

Dialogic usually finds a stack size of 64K to be sufficient but, depending on the application stack allocation demands, more memory might be required.

Using BOSTDLLD with Debugging Enabled

When using the DLL version of the Bfv API library with a debug version of your application, take special care when handling FILE * file pointers or dynamically allocated memory.

Some functions use FILE * pointers that the application must open. Applications can also access FILE * pointers that the Bfv API has opened (with the `TIFF_FP` macro). There are also some instances where memory is allocated by the Bfv API that must be freed by the application, for example, ***BfvLinesAvail*** and `PAGE_RES` structures.

The runtime libraries used by BOSTDLLD are separate from the debug ones used by the application. The meanings and applicability of the file pointers and allocated memory differ and are incompatible between the two copies of the runtime libraries. To avoid these kinds of pointer problems, the Bfv API provides a set of ***_dll_...*** functions. See ***Volume 1*** of the ***Bfv API Reference Manual.***

Determining Installation Parameters

Any time after the driver has been installed (but not removed), you can determine what installation parameters were used.

➢ ***To determine installation parameters. do the following:***

1. Run ***regedit*** or ***regedit32***.
2. Locate the key:

   ```
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
   \Boston\Parameters
   ```
3. Look at the values under this key.

Driver Startup

After you install the driver, it has a service associated with it named ***boston***, a kernel-mode driver.

If driver startup fails, the error message that appears is almost always unrelated to the actual cause of failure. The Event Log contains an error message supplied by the driver. Using the Event Viewer, look for the most recent entries in the System section labeled ***boston***. Occasionally the Event Log fills up. If this happens, set ***Action \ Properties \ Overwrite*** events as needed.

Under kernel debugging environments, the debugger also contains a message. Under other conditions, entries can also appear in the ***Event Log \ debugger***. If you encounter unusual behavior, check the ***Event Log \ debugger***.

Global Variables

If you plan to compile your application for use with either DLL version of the Dialogic® Brooktrout® Bfv API library, do not use local variable names that match any of the names of the global variables defined by the Bfv API library. Such duplications will prevent the application from compiling.

The Bfv API global variables include:

- the variables used by ***getopt*** (for example, `optind`)
- the `_mill_priority` variable, specific to Windows®

Limits on Open Files

The Microsoft® Visual C++® compiler library limits the number of files that can be opened. Developers often encounter this limit when developing multithreaded Dialogic® Brooktrout® Bfv API applications. Microsoft has updated their libraries, greatly increasing this limit. For Visual C++® 4.1+, the limit is 512 files. You can increase this limit using the ***_setmaxstdio*** function from the compiler library.

Process Priority

The Bfv API automatically raises the priority of any process that calls ***BfvLineAttach*** to `HIGH_PRIORITY_CLASS`. Dialogic recommends using this priority to keep delays to application processes performing fax or speech functions to a minimum. Priority raising occurs the first time the application calls ***BfvLineAttach*** only, and is controlled by the global variable `_mill_priority`, which contains the new priority for the process.

Command Line Configuration for *Install*

When configuring the device driver using ***install*** or removing it using ***install -r***, you can specify the command line option `-c` to cause the ***install*** program to not prompt you for any parameters. Instead, the ***install*** program uses the responses that you supply following the `-c` option on the command line.

If you use the `-c` option, you must include all the parameters described below, in order, regardless of whether or not the parameter has meaning on the system in question. If a parameter does not apply, it is ignored. The parameters are not checked for validity.

Note: The software does not prompt for removal of old driver versions.

When configuring the driver using ***install***, you must include the following parameters:

Parameter	Defaults
Reserved	p
Reserved	0
Reserved	-1
Max num PCI hardware modules	16
Physical buffer size	32768
Application buffer size	10240
Reserved	0
Machine ID	1
Reset option (1 or 0)	1
History Enable	0

History Size	1024000
Num physical histories	0
Num app histories	0
Restrict phys channels (1 or 0)	1
Number of I2O frames	*0
Flow control interval	*0
Memory alloc minimum	*0
Memory alloc quanta	*0
Reserved	0
Test parameter 1	*0
Test parameter 2	*0
Test parameter 3	*0
Reserved module info module id, module number 0 to terminate	*0
Load driver with no devices present (y or n) - (n = pnp, y = software only solution)	n
Do mem adjustments (y or n)	n
What to do if already adjusted (1, 2, or 3)	1
Memory in system	0
Reserved	0
Reboot after mem adjustment (1 or 0)	0

The advanced parameters, marked with * in the list above, can always be safely set to 0. This setting tells the driver to use the default values.

For example, to configure the driver for up to 16 PCI hardware modules use the following settings:

physical buffer size 32768
application buffer size 10240
machine ID 1
reset on
history enabled
history size 1024000
physical histories 0
application histories 0
restrict physical channels 0
all default advanced parameters 0

not to start at boot 0
no memory adjustments 0
not reboot 0

Enter the following at the command line:

```
install -c p 0 -1 16 32768 10240 0 1 1 1 1024000 0 0 1 0 0 0 0 0 0 0 0 0 n n
     1 0 0 0
```

For more information, see ***Windows® Operating System on page 278***.

Driver File Locations (install)

When using ***install*** to configure the device driver, you do not need all files from the distribution. However, the file locations used by the configuration procedure relative to the directory where ***install*** is run must be the same as that in the normal distribution.

Assuming a directory named ***install*** where ***install*** is being run, the following files must be present:

install\install.exe

pnp\brooktrout.cat
pnp\trxstream.inf
pnp\TRxStream\x86 (directory for the following 32-bit OS files)
- ***boston.pdb***
- ***boston.sys***
- ***brktBdevco.dll***
- ***brktBdevpp.dll***

pnp\TRxStream\x64 (directory for the following 64-bit OS files)
- ***boston.pdb***
- ***boston.sys***
- ***brktBdevco.dll***
- ***brktBdevpp.dll***

These files and directories appear in the standard distribution under the directory ***driver\winnt***.

Build Requirements for install

You need the Windows® 2000 DDK installed to compile the ***install.exe*** application. For more information about the Microsoft® Windows® Device Driver Kit, see ***microsoft.com***. ***install.exe*** links to the DDK libraries: ***setupapi.lib*** and ***newdev.lib.*** It also links to the ***shlwapi.lib*** system library.

Version Information

The driver file, ***boston.sys***, and the DLL library file, ***bostdlld.dll***, contain version resource information. Use the system File Manager or Explorer to view this information.

Reviewing Compiler and Linker Options

The makefiles contained in the ***app.src*** and ***bapp.src*** directories provide a number of command line options to the compiler and linker. Using these options, the compiler and linker produce object files and executables compatible with the Bfv API library.

Dialogic strongly recommends that you use these options for all your Bfv API applications. But for those who want to change the set of options the compiler and linker use, the following is a list of the nonstandard options and their functions. The options used for compilation are marked with (***C***) and those for linking are marked with (***L***).

The makefile uses the standard `$(cflags)` from ***ntwin32.mak*** and `$(cvarsdll)`for the dynamically-linked C runtime libraries. It uses the standard `$(conlflags)` for linking.

`$(cflags)`	(***C***) Options for CFLAGS.
$(cvarsdll)	(***C***) Options defining dynamic runtime library.
$(conlflags)	(***L***) Options for linking console mode program.
`-nologo`	(***C***) Do not print logos.
`-DWINNT`	(***C***) Defines a symbol that identifies the environment.
`-UMSDOS,-U_MSDOS`	(***C***) Undefines symbols that the compiler might define that can cause problems at compilation.
`-DMILL_DLL`	(***C***) Defines a symbol that is required when using the DLL Bfv API library.
`libcpmt.lib`	(***L***) C++ libraries. Required when new call control is used.
`NODEBUG=1`	**Note:** This is a makefile line, not a compilation or link flag. This line is needed in the makefile when compiling with certain versions of Visual Studio® to prevent it from automatically compiling for debugging and producing inappropriate warning messages.

6 - Configuring Software

This chapter describes how one can configure the Dialogic® Brooktrout® software for your Dialogic® Brooktrout® boards.

Dialogic has created ways to configure the Dialogic® Brooktrout® software including:

- ***Using the Dialogic® Brooktrout® Configuration Tool on Windows Systems on page 91***
- ***Running the Configuration Tool on page 94***

  For Windows® operating systems, you can use the Dialogic® Brooktrout® Configuration Tool in one of several modes:

  - ***Wizard Mode on page 104***
  - ***Advanced Mode on page 110***
  - ***Offline Mode on page 137***
  - ***Silent Mode on page 142***

  When a Configuration Tool parameter panel includes Advanced Settings, use the Advanced Settings button to view them.

  - ***Advanced Settings on page 143***
- ***Using a Call Control Configuration File on page 145***

  For all operating systems, you can configure the call control by creating a call control configuration file that is read by the ***BfvLineReset*** function.

For more information, see:

- ***Volume 6, Appendix A*** in the ***Bfv API Reference Manual***
- Bfv API Function Locator in any volume of the Reference Manual

Using the Dialogic® Brooktrout® Configuration Tool on Windows Systems

When you use one of the Windows® operating systems, you can configure your module with the Dialogic® Brooktrout® Configuration Tool for Windows® (***configtool.exe***).

Note: For the Dialogic® Brooktrout® Configuration Tool to work with Windows® 2000, you must install Internet Explorer 6.0 or higher.

The Dialogic® Brooktrout® Configuration Tool is a graphical user interface program that you can use to create and modify the configuration files, edit and update the driver parameter, and configure and initialize the module. The Configuration Tool also installs the Plug-and-Play driver, if not already installed.

It can be launched in several modes. You can find additional details for these modes at the following locations:

- ***Wizard Mode on page 104***
- ***Advanced Mode on page 110***
- ***Silent Mode on page 142***
- ***Timer Mode on page 93***

A brief summary of each mode function is highlighted here:

Wizard Mode

Guides you in configuring Computer Telephony and software. To use Wizard mode, all Dialogic® Brooktrout® modules in your system must be of the same type and the configuration tool must recognize the modules as Dialogic® Brooktrout® boards.

When the system contains different types of modules or the Configuration Tool does not recognize the modules, the Configuration Tool selects Advanced Mode automatically.

See ***Wizard Mode on page 104***.

Advanced Mode

Requires that you have some knowledge of computer telephony, communications protocols, and Computer Telephony. The Configuration Tool automatically selects Advanced Mode when there are different types of modules installed or when you explicitly indicate Advanced Mode by using a command line switch to start, possibly to configure each module differently (even if the modules are the same kind).

A toolbar provides the following options:

- The Save button:

  Saves the configuration.
- The Apply button:
  - Saves your configuration to the appropriate configuration file.
  - If driver parameters were modified, it dynamically reconfigures the drivers on a PnP operating system.
  - If the Boston Host Service is not installed, installs the Boston Host Service.
  - If Boston Host Service is not running, stops the Boston Host Service.
  - Starts the Boston Host Service.

Offline Mode

Allows you to create/edit configuration files when no boards are installed. In Offline Mode, the toolbar option Apply is not present and no checks are performed on the existence of the modules specified in the configuration.

The toolbar enables two options that are disabled in Advanced Mode: Add and Remove. These buttons allow you to modify the configuration files to add or remove modules.

See ***Offline Mode on page 137***.

Silent Mode

Reads configuration files from the location specified at installation (usually ***C:\Brooktrout\Boston\config)***, starts the device driver (if necessary). The graphical user interface is not displayed.

The configuration tool creates a log file (***configuration.log***) that contains all the actions that it executed and the results of these actions. This log file is located in the current working directory.

See ***Silent Mode on page 142***.

Timer Mode

Sets a timer that launches the Configuration Tool after a specified amount of time. You can override the time and launch the Configuration Tool immediately. You might want to override the timer when you do not need to check on the existence of modules.

Running the Configuration Tool

You can run the Configuration Tool from:

- The program menu:

  `Start → Program → Brooktrout → Configuration Tool`

- The command line:

  `C:\Brooktrout\Boston\utils\winnt\bin\configtool.exe`

  - To specify the mode on the command line, use one of the following:

    ```
    /a – Advanced Mode
    /o – Offline Mode
    /s – Silent Mode
    /t – Timer Mode
    /w – Wizard Mode
    ```

  - Other command line arguments include:

    `/c (configip)`

    Displays the following note:

    Note: `/c` option brings up the Configure IP Stack dialog box only if there are IP enabled devices installed.

    `/d (debug)`

    Enable the debug flag, which initiates writing of the debug information to the ***configuration.log*** file.

    `/e (expert)`

    Show all hidden parameters. Use this mode only under the guidance of Technical Support personnel. See ***Getting Technical Support on page xxxvi***.

    `/i <INF file location>`

    Use the specified INF file to install the PnP driver when it is not installed or installed but not started.

    You must uninstall any previous version of the plug and play driver INF file. See ***Removing the Plug-and-Play Driver on page 68*** or ***Removing the Driver with a Command (cmd) Prompt on page 76***.

`/f <cfgfile>`

Specify the name and location of the user-defined configuration file <cfgfile> while launching the Configuration Tool. For example:

`/f C:\Brooktrout\Boston\custom\btcall.cfg`

`/p`

Show the Brooktrout Configuration Tool - Preferences dialog box

`/q`

Do not display the Brooktrout Configuration Tool - Preferences dialog box.

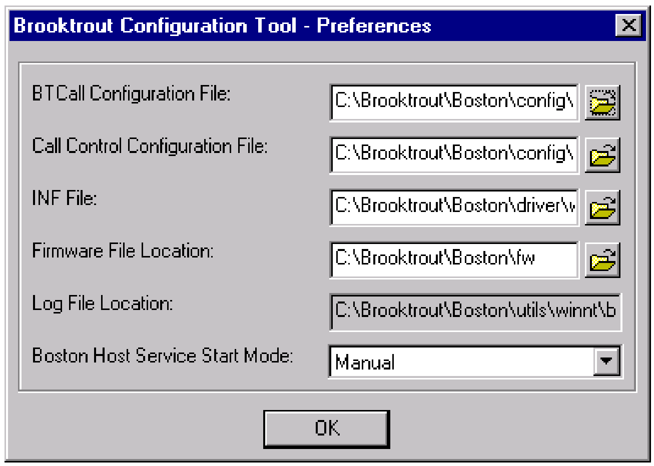
You can use any options in combination with a mode when launching the Configuration Tool. When run from any mode, including the command line without a mode option switch, the Configuration Tool starts in Wizard Mode, if the modules are of same type. Otherwise, the Configuration Tool starts in Advanced Mode.

Choosing the File Location

When you start the Configuration Tool for the first time, you can choose the locations for various files using the Brooktrout Configuration Tool - Preferences dialog box.

Note: You might want to set these file locations before you start the Configuration Tool. See ***Setting File Locations Prior to First Execution on page 96***.

Preferences Dialog Box



Setting File Locations Prior to First Execution

➢ **To suppress the Preference dialog box with specified configuration files locations other than the defaults, before the first use of the application, follow these steps:**

1. Modify the following fields in the ***settings.cfg*** file (see ***Example of settings.cfg File on page 97***):
   - `btcall_cfg_path`
   - `cc_cfg_path`
   - `log_cfg_path`
   - `firmware_dir`

   The ***settings.cfg*** file must be placed in the same directory as the Configuration Tool program (***configtool.exe***):

   `C:\Brooktrout\Boston\utils\winnt\bin`

2. Launch the Configuration Tool with the /q command line option:

C:\Brooktrout\Boston\utils\winnt\bin\configtool.exe /q

After launching the Configuration Tool for the first time, you can access the Preferences dialog box only through Advanced Mode (see ***Advanced Mode Buttons on page 113***) or by launching the Configuration Tool using /p command line option:

C:\Brooktrout\Boston\utils\winnt\bin\configtool.exe /p

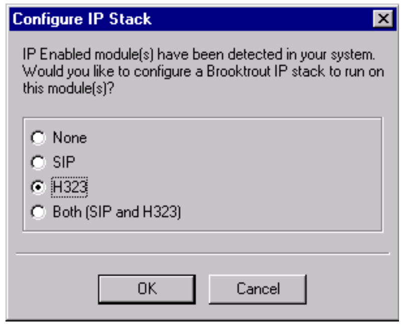
Example of settings.cfg File

```
ip_stack_config=sip
home_dir=C:\Brooktrout\Boston
boston_source_path=C:\Brooktrout\Boston\driver\winnt\kernel\compile\i386
   \boston.sys
inf_source_path=C:\Brooktrout\Boston\driver\winnt\pnp\trxstream.inf
bostsrv_source_path=C:\Brooktrout\Boston\bfv.api\winnt\bin\bostsrv.exe
firmware_dir=C:\Brooktrout\Boston\fw
protocol_dir=C:\Brooktrout\Boston\config
btcall_cfg_path=C:\Brooktrout\Boston\config\btcall.cfg
cc_cfg_path=C:\Brooktrout\Boston\config\callctrl.cfg
log_cfg_path=C:\Brooktrout\Boston\bin\configuration.log
lic_mgr_path=C:\Brooktrout\Boston\bin\brktlicmgr.exe
modules_are_connected=false
vb_fname=C:\Brooktrout\Boston\fw\bostvb.dll
```

Configure IP Stack Dialog Box

When the Configuration Tool detects an IP-enabled device for the first time, it displays the Configure IP Stack dialog box:

Choose the IP stack you want to configure:



- None: IP stack not installed
- SIP: SIP stack installed (default)
- H.323: H.323 stack installed
- Both (SIP and H.323): Both SIP and H.323 stacks installed

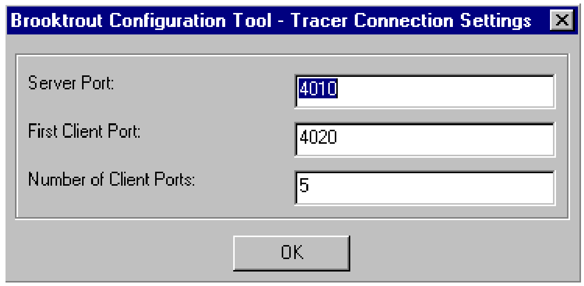
When any option is selected, all previously added host modules are removed.

The settings for the IP stack can also be made in the ***settings.cfg*** file, see ***Example of settings.cfg File on page 97***.

Trace Connection Settings Dialog Box

You use the Trace Connection Setting dialog to configure the connection parameters for the Brktcctrace (server) application, and your end user application (client). See the ***Debugging*** chapter in your developer guide for details about the ***Brktcctrace*** application.

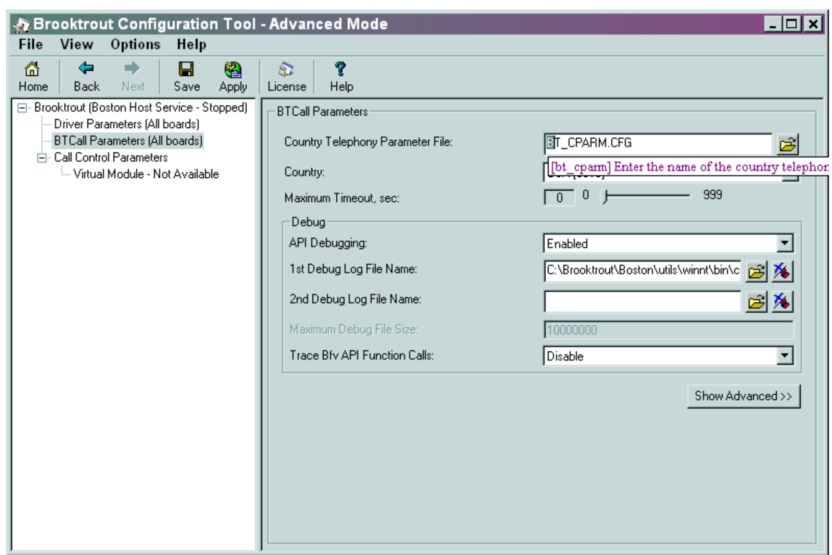
Server Port:	Specifies the server port number.
First Client Port:	Specifies the client port that the server will first try to connect to.
Number of Client Ports:	Specifies the maximum number of clients.



Getting Help From the Configuration Tool

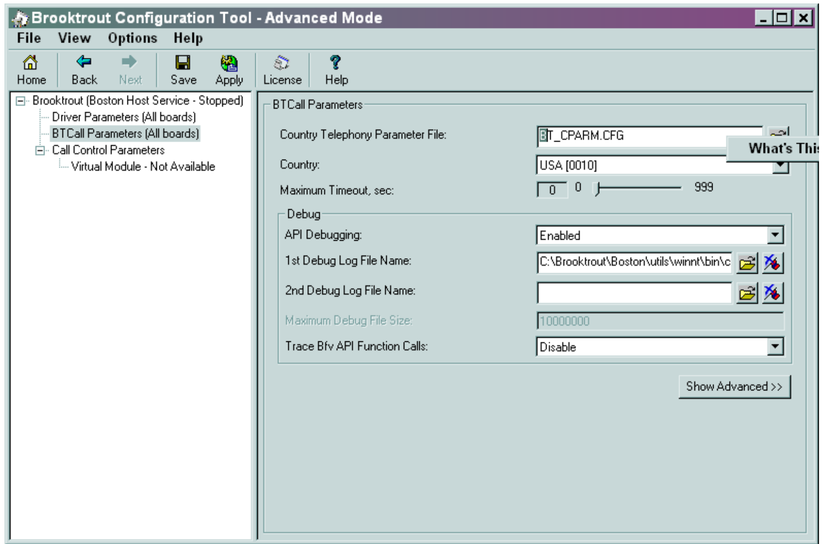
The Configuration Tool provides context sensitive online help for parameters. The Help displays the purpose and description of the parameter including the default value and range, if applicable.

- Hold the cursor over the parameter input area to see a brief parameter definition:

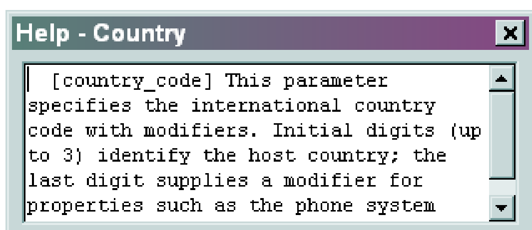


- Right-click the parameter drop-down box to get a more complete definition for Configuration Tool parameters.

  The What's This? popup menu appears:



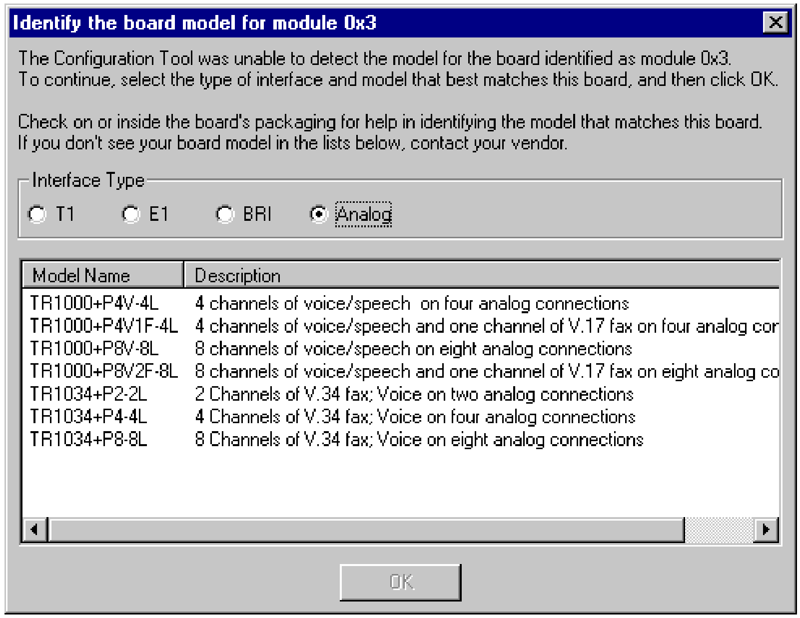
- Left-click the What's This? popup menu for a definition like the following. You can resize this text box, if necessary.



Identifying the Model of the Board

➢ **When you launch the Configuration Tool and it is unable to automatically identify the board:**

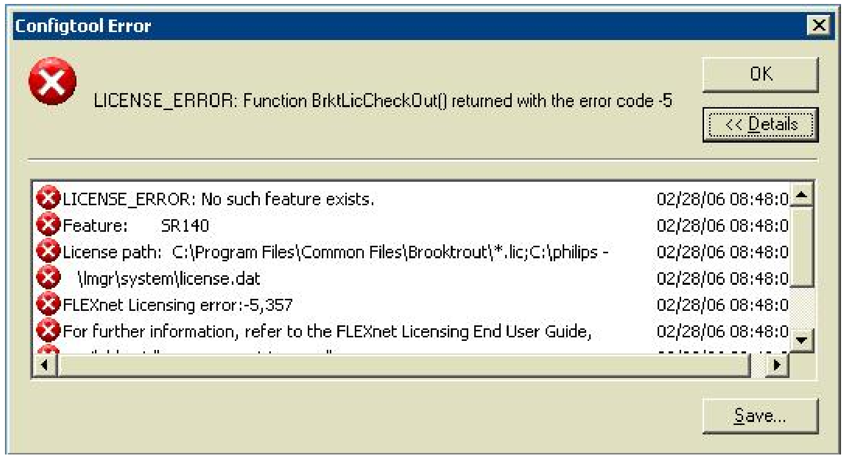
1. The Configuration Tool displays a dialog box similar to the following.



2. Choose an Interface Type and highlight the board model, then click OK.
3. If the Configuration Tool is not able to detect the board, then it launches the tool in Advanced Mode, (see ***Advanced Mode Buttons on page 113***) even if you included the `/w` parameter on the command line.

Resolving the Configtool Error Message

When you launch the Configuration Tool and the tool displays the Configtool Error screen, use the following instructions to eliminate this licensing conflict.



This error message displays when you have licenses from other vendors installed in a generic location and not in a vendor specific location. This conflict is possible with other products that use Macrovision licensing, and occurs when a Macrovision-protected application uses the `LM_LICENSE_FILE` environment variable rather than a vendor-specific name.

The Macrovision FlexNET licensing module queries for SR140 licenses. The query fails for the licenses that are installed in the generic location due to absence of the necessary vendor information.

This error message can only occur when a SR140 Host Based Fax module license is not installed.

➢ **To resolve this conflict and eliminate the Configuration tool error message screen:**

1. On the Desktop, right-click on My Computer
2. Select:

   Properties → Advanced → Environment Variables
3. Click New and create an environment variable <VENDOR_NAME>_LICENSE_FILE, replacing <VENDOR_NAME> with the application or vendor name. For example, `ACME_DLS_LICENSE_FILE`
4. Set the value of the <VENDOR_NAME>_LICENSE_FILE variable to the same value as `LM_LICENSE_FILE`.

   For example:

   If `LM_LICENSE_FILE = C:\acme\lmgr\system\license.dat`, then set

   ```
   ACME_DLS_LICENSE_FILE =
   C:\acme\lmgr\system\license.dat
   ```
5. To delete the `LM_LICENSE_FILE` environment variable, highlight the current value with the mouse and click on Delete.
6. Click Start → Run.
7. Enter **`regedit`**, and click OK.
8. Expand:

   `HKEY_LOCAL_MACHINE\SOFTWARE\FLEXlm License Manager` key and verify that there is a registry entry named `LM_LICENSE_FILE` under this key.
9. If the `LM_LICENSE_FILE` registry entry exists, rename the entry to **<VENDOR_NAME>_LM_LICENSE_FILE**. Replace <VENDOR_NAME> with the application or vendor name. For example, `ACME_DLS_LICENSE_FILE`
10. Verify:
    a. The Dialogic® Brooktrout® Configuration Tool no longer displays the error message box.
    b. The other application using Macrovision licensing still runs.

Wizard Mode

In Wizard Mode, after you answer the basic questions, the Configuration Tool:

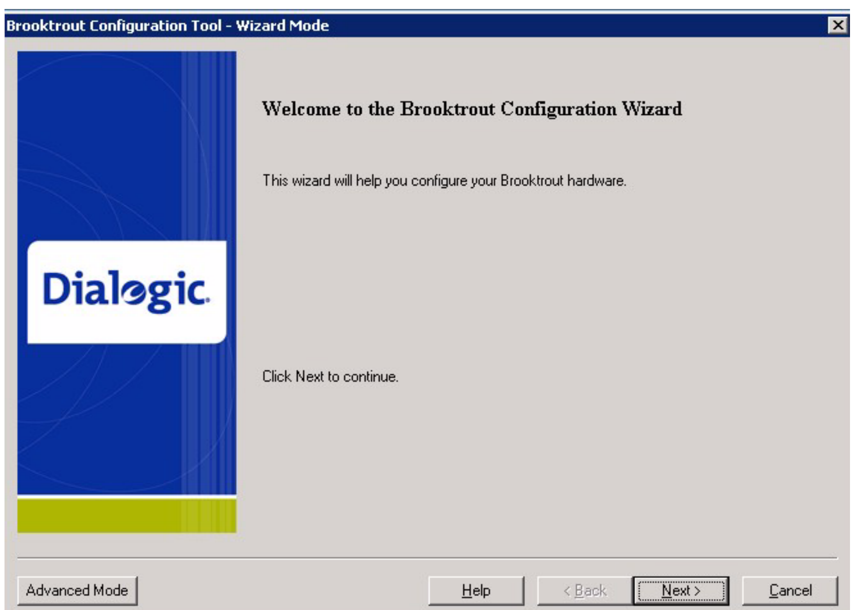
- Automatically creates the necessary configuration file
- Starts the drivers, if required
- Starts the Boston Host Service

Wizard Mode requires that all the modules be of the same kind. All modules are configured identically, that is, the modifications to the configuration apply uniformly to all the modules.

Using Wizard Mode

➢ **To use Wizard Mode:**

1. Start the Configuration Tool if it has not already been started by the Plug-and-Play installation procedure. See ***Running the Configuration Tool on page 94***.
2. The Welcome window appears.



3. When the Welcome window appears, you can choose to click the Advanced Mode button or the Next button.

   Choose one of:

   - Advanced Mode

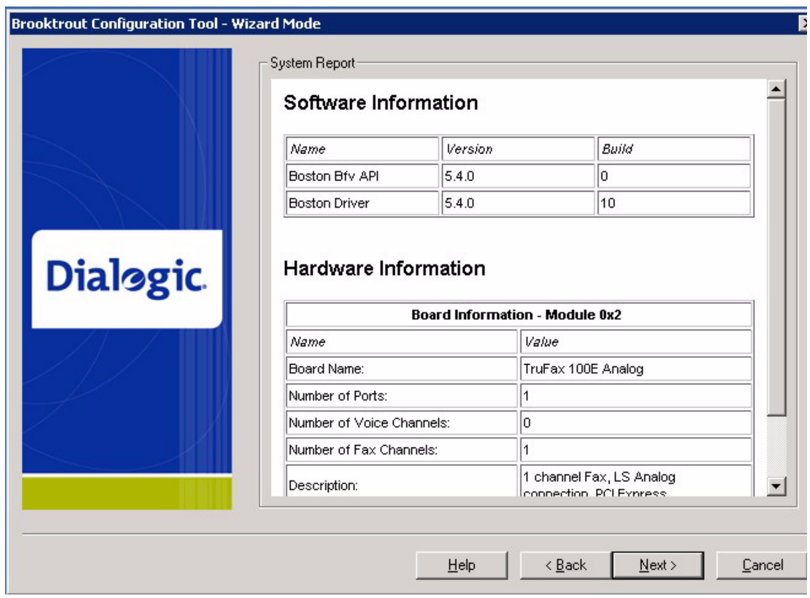
     If you select Advanced Mode, the Configuration Tool exits Wizard Mode and starts in Advanced Mode (see ***Advanced Mode on page 110***).

   - Next

     If you click the Next button, Wizard Mode continues and creates the necessary configuration file after you answer the basic questions.

System Report

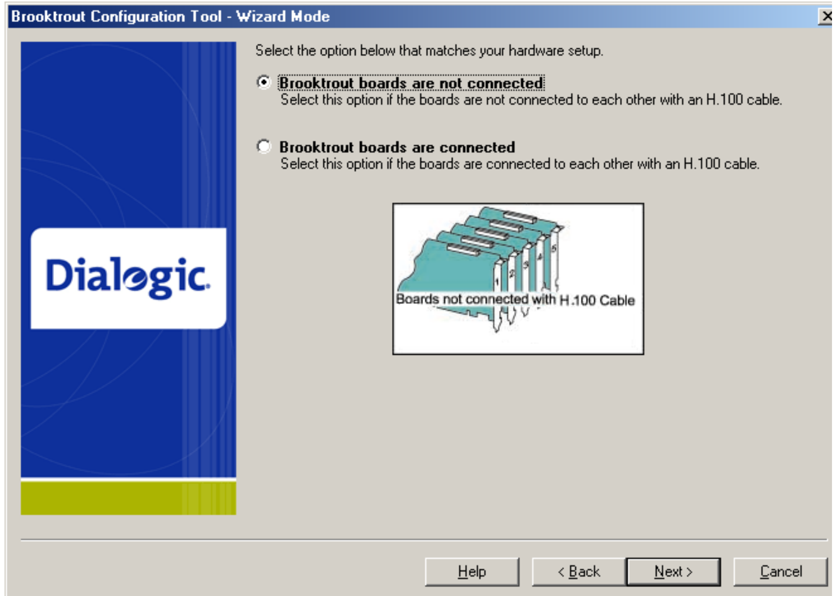
4. The System Report appears on the right-hand panel, then click Next.



This example shows the information for a T1 board.

Board Connection Information

5. The next dialog box appears when you have more than one board with an H.100 connector.



The configuration tool cannot detect if the boards are interconnected, so you must select one of the radio buttons. Depending on the radio button selection, the configuration tool sets the appropriate defaults for the clock settings.

Choose one of:

- Brooktrout boards are not connected (default)

  Clock Mode is Master and all the boards configure as master.

- Brooktrout boards are connected

  Clock Mode: The first detected module is configured as master and the remaining modules configure as slaves.

  master_drive=clock_a

  See the Call Control Configuration File in ***Volume 6*** of the ***Bfv API Reference Manual*** for a description of setting clock modes and sources.

Note: SR140 modules are not detected in Wizard Mode.

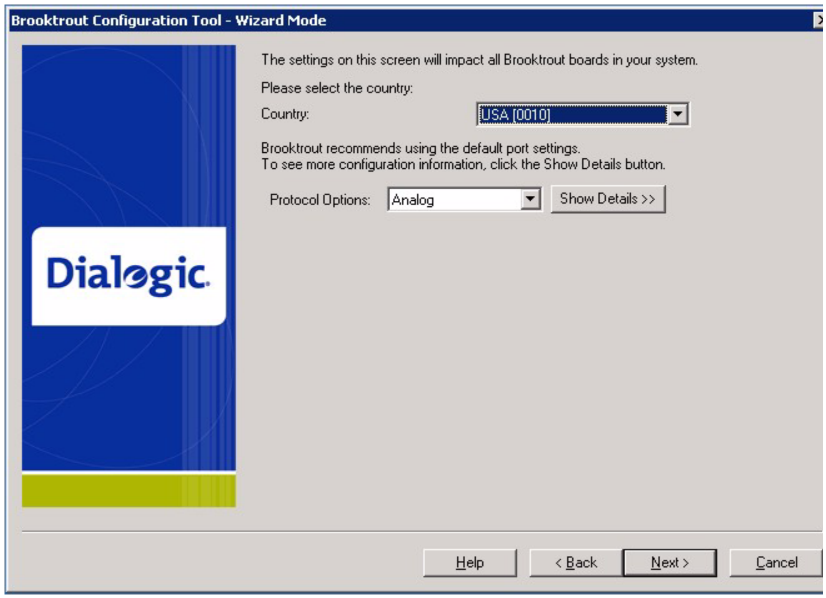
6. Click **Next**.

   The **Protocol Options** dialog box appears.

Protocol Options

Dialogic recommends that you accept the defaults that are based on the modules the configuration tool identified in your system. In this example, the module uses the T1 ISDN protocol.

Note: In Wizard Mode all the ports of all the modules will be configured with the same port setting.



7. Click the Show Details >> button.

   Note the details of the protocol chosen by the Configuration Tool.

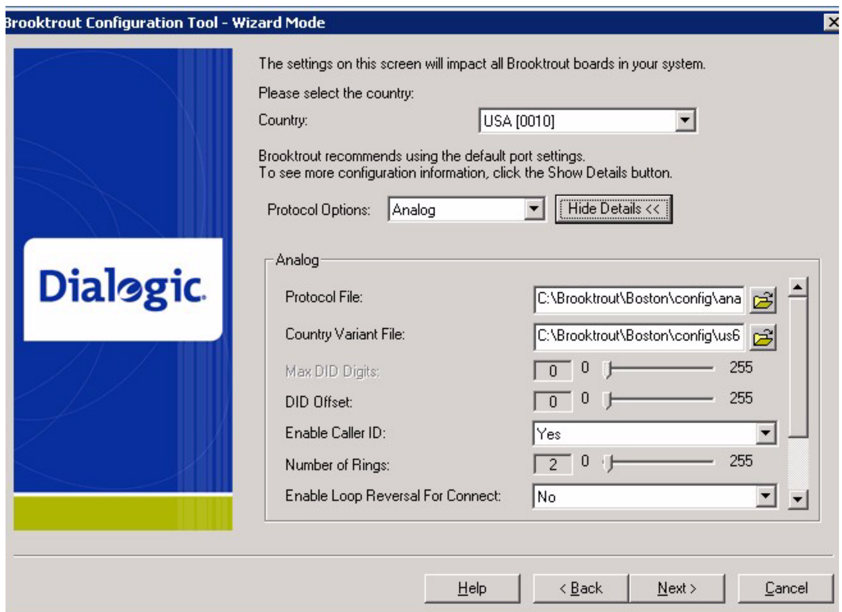
Protocol Details

The protocol details appear in the panel below the Protocol Options. The options are dependent on the protocol chosen.

This example displays the options for the T1 ISDN protocol.

See the Configuration Files in ***Volume 6*** of the ***Bfv API Reference Manual*** for more information about the protocol parameters.

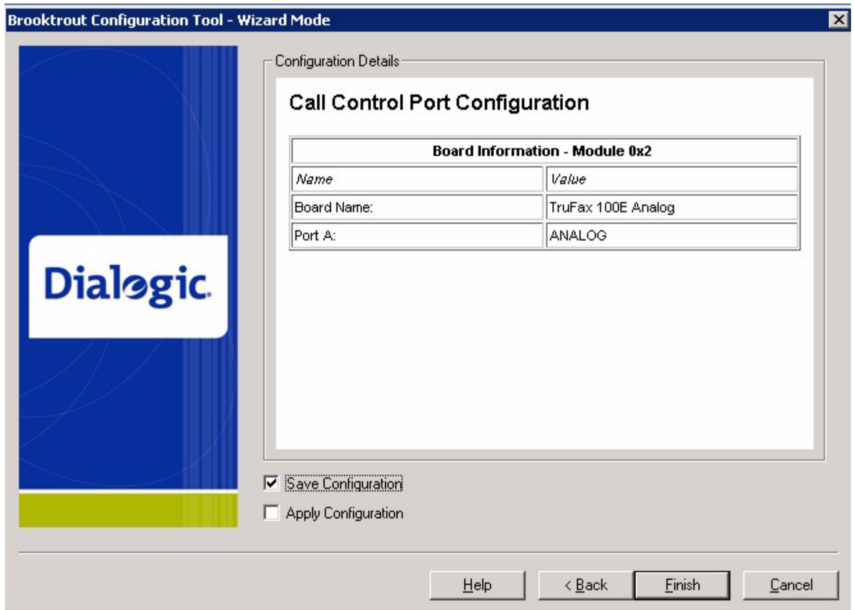
Note: In wizard mode all the ports of all the modules will be configured with the same port setting.



8. Click Next.

Configuration Details Panel

The Configuration Details panel appears.



9. Check either or both options:

   - Save Configuration

     Saves the configuration to the configuration files. If you select Save Configuration, the Next button changes to Finish.

   - Apply Configuration

     Saves the configuration and downloads the firmware for all the detected modules.

     If you select Apply Configuration, the Configuration Tool applies the configuration and then the Wizard Mode Finish window appears.

10. Click Finish or Next depending on your selection.

    The Finish window appears if you clicked Next.

11. Click Finish to complete the configuration.

Advanced Mode

If you have different types of Dialogic® Brooktrout® modules in your system or you want to configure your identical modules differently, you must use Advanced Mode:

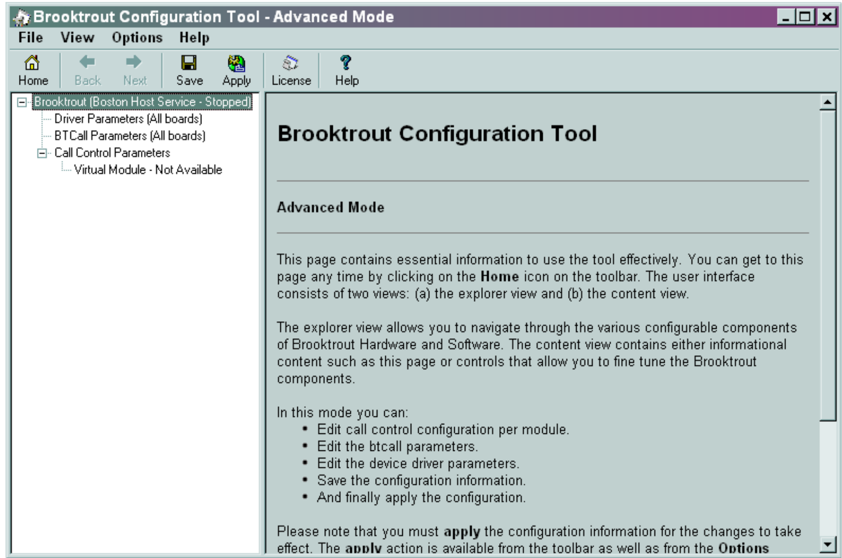
- When the Dialogic® Brooktrout® Configuration Tool detects that you have different modules, it starts in Advanced Mode.
- When you have IP-enabled modules, the Dialogic® Brooktrout® Configuration Tool always starts in Advanced Mode.
- When you start the Configuration Tool in Wizard Mode, you can move to Advanced Mode immediately by clicking the Advanced Mode button.

The Advanced Settings parameters for all module types are presented in logical groups. The ordering of the groups is alphabetical. The ordering of parameters within a group is according to their relative importance.

Most parameters are hidden unless you click Show Advanced. Generally, a user does not need to change the advanced parameters. See ***Call Control Parameters on page 118***.

Advanced Mode Window

When you enter Advanced Mode, the Brooktrout Configuration Mode - Advanced Mode window appears.


Brooktrout Configuration Tool - Advanced Mode
File View Options Help
Home Back Next Save Apply License Help
Brooktrout (Boston Host Service - Stopped)
Driver Parameters (All boards)
BTCall Parameters (All boards)
Call Control Parameters
Virtual Module - Not Available
Brooktrout Configuration Tool
Advanced Mode
This page contains essential information to use the tool effectively. You can get to this page any time by clicking on the Home icon on the toolbar. The user interface consists of two views: (a) the explorer view and (b) the content view.
The explorer view allows you to navigate through the various configurable components of Brooktrout Hardware and Software. The content view contains either informational content such as this page or controls that allow you to fine tune the Brooktrout components.
In this mode you can:
• Edit call control configuration per module.
• Edit the btcall parameters.
• Edit the device driver parameters.
• Save the configuration information.
• And finally apply the configuration.
Please note that you must apply the configuration information for the changes to take effect. The apply action is available from the toolbar as well as from the Options

Navigating in Advanced Mode

➢ **To navigate in Advanced Mode use the menus and buttons on the tool bar and the tree view on the left panel of the dialog boxes.**

Select one of the following to view the accompanying right panel information:

- Select Brooktrout to display the welcome window, if you are not already there.
- Select Driver Parameters to display the Driver Parameters dialog box.

  For sample values, see ***Driver Parameters (All boards) on page 114***.
- Select BTCall Parameters to display the BTCall Parameters dialog box.

  For sample values, see ***BTCall Parameters (All boards) on page 116***.
- Select Call Control Parameters to display the Call Control Parameters dialog box.

  For sample values, see ***Call Control Parameters on page 118***.
- Select IP Call Control Parameters to display the IP Call Control Parameters dialog box.

  For sample values, see ***IP Call Control Modules on page 130***.

Advanced Mode Menus

➢ **The following menu items are available in the Advanced Mode window:**

- The File menu allows you to exit from the configuration tool.
- The View menu allows you to choose the previous or next window.
- The Options menu allows you to:
  - Choose to apply and/or save the configuration.
  - Reconfigure the driver.
  - Launch the Dialogic® Brooktrout® License Manager.
  - Configure the IP Stack.
  - Edit Trace Connections Settings.
  - Set your preferences for the locations of the ***btcall.cfg*** and ***callctrl.cfg*** configuration files, the firmware, and log files.

    The Preferences dialog box is shown in ***Choosing the File Location on page 96***.
  - Display the toolbar.

- The Help menu provides the following options:
  - Access to the Configuration Tool Help Index Contents tree.
  - Access to the Configuration Tool Help Index tab where you can navigate through the Help index entries.
  - Search tab where you can look for any text entry.
  - About provides version and copyright information for the Dialogic® Brooktrout® Configuration Tool.

Advanced Mode Buttons

➢ **The Advanced Mode window contains these buttons:**

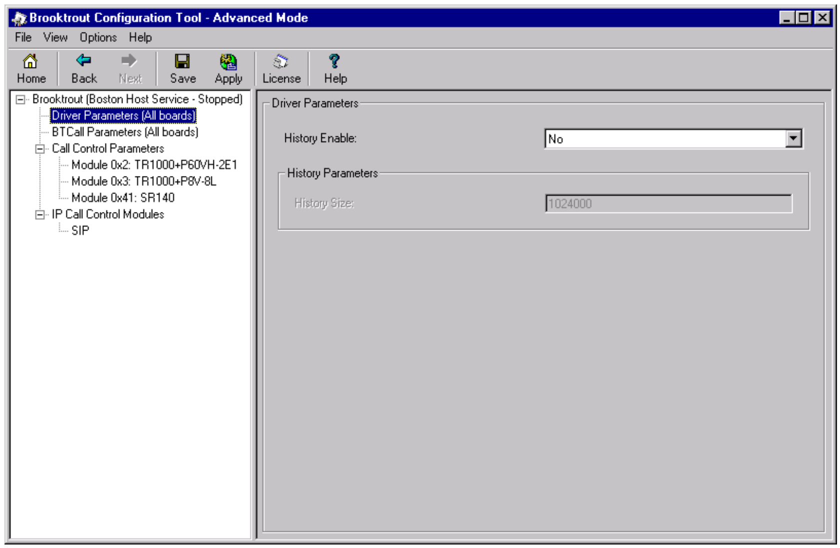
- Select the Home button to go to the welcome window if you are not already there.
- The Next and Back buttons take you to the next or previous pages, respectively.
- The Save button saves your configuration.
- The Apply button installs, starts or stops the Boston Host Service. It also saves your configuration to the appropriate configuration file.

  If driver parameters were modified, it dynamically reconfigures the drivers on a PnP operating system.
- The License button launches the Dialogic® Brooktrout® License Manager (***bkrtlicmgr.exe).***
- The Help button displays help for the Configuration Tool in any mode.

Driver Parameters (All boards)

Click on Driver Parameters (All Boards) in the left panel to display the Driver Parameters in the right panel.

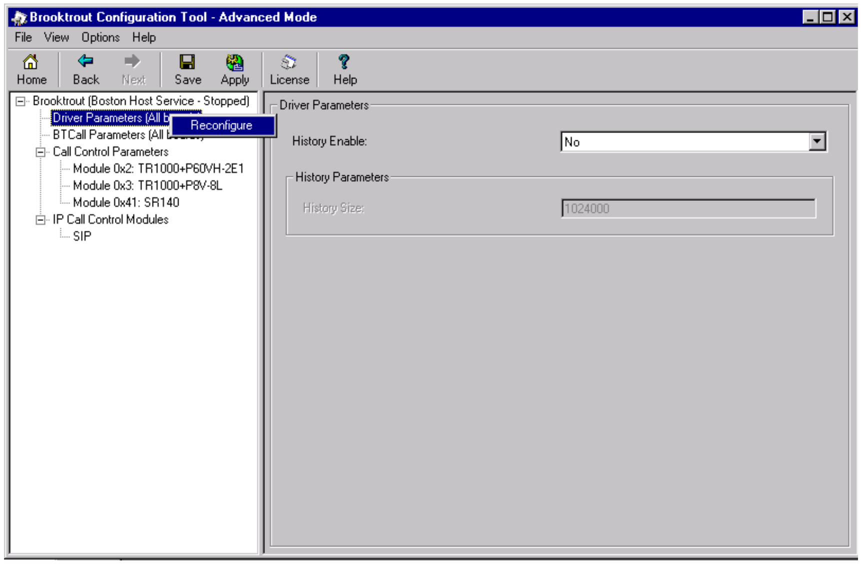
See ***Installing and Configuring the Driver using a Command (cmd) Prompt on page 72***.



Reconfigure Drivers

Right-click on Driver Parameters (All Boards) in the left panel to reconfigure the driver.

This option dynamically updates the driver parameters by calling ***reconfig*** on a PnP system.



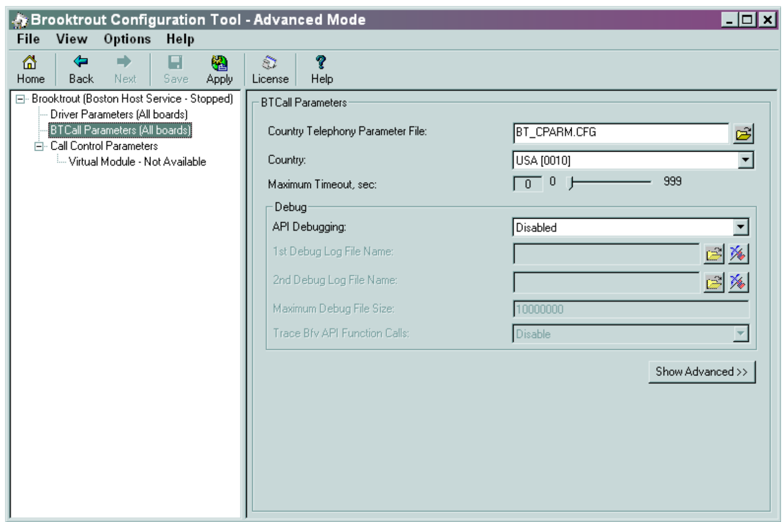
BTCall Parameters (All boards)

Click on BTCall Parameters (All Boards) in the left panel to display the BTCall Parameters in the right panel.

Debugging Options

Debugging Disabled

When debugging is disabled, all the BTCall Parameters debugging-related parameters are grayed out.



See the User-Defined Configuration File in ***Volume 6*** of the ***Bfv API Reference Manual*** for parameter information.

For more information about the Show Advanced >> button, see ***Advanced Settings on page 143***.

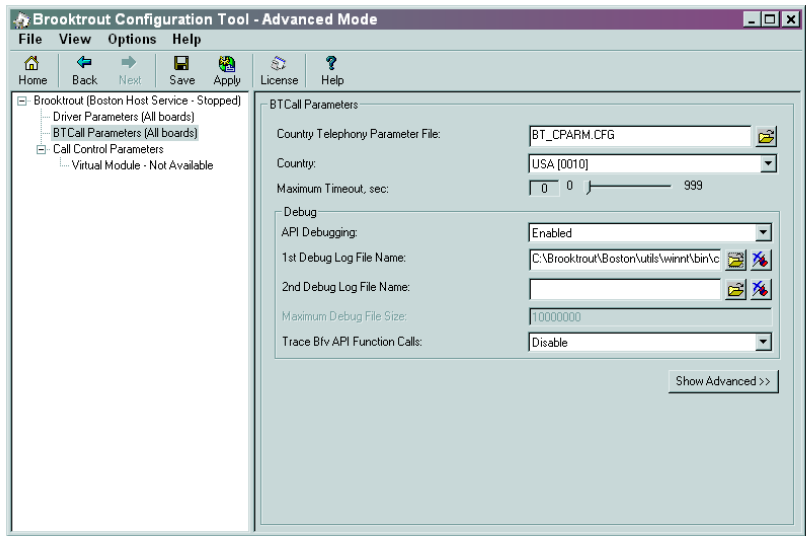
Note: Most users never need to alter the Advanced Settings parameter values.

Debugging Enabled

When you enable API Debugging, you can:

- Trace the entry
- Exit to each Bfv API call
- List the parameter passed into a function

See the right panel for specific debug parameter information.



Call Control Parameters

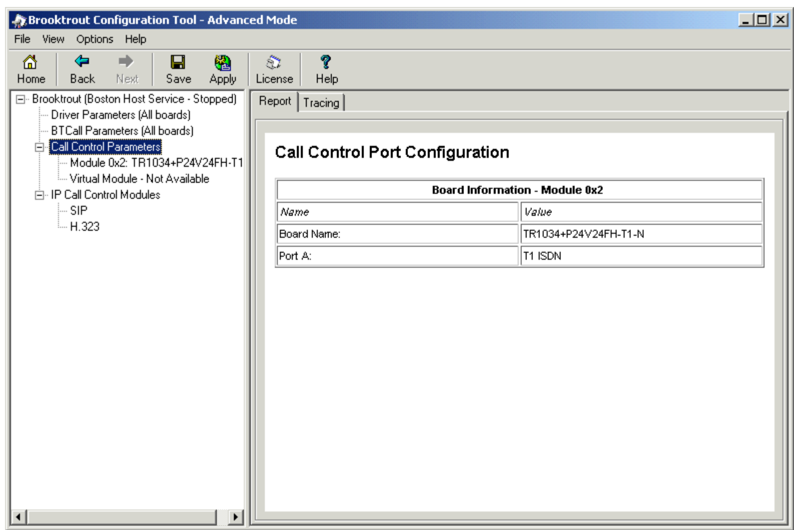
Click on Call Control Parameters in the left panel to display the Report tab in the right panel.

See ***About the Call Control Configuration File on page 145*** and ***Volume 6*** of the ***Bfv API Reference Manual*** for parameter information.

Reporting and Debugging (Tracing)

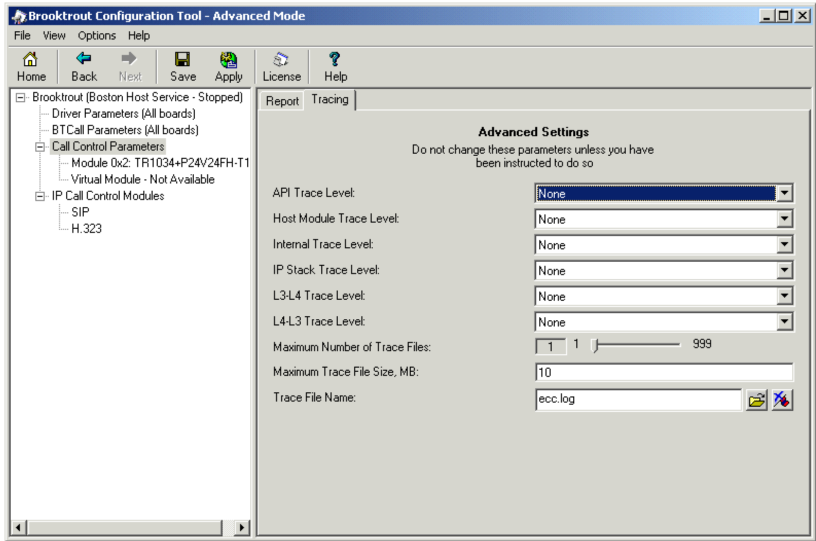
Report Panel

The Report tab displays the Call Control Parameters port configuration information.



Tracing Panel

The Tracing tab displays the Call Control Tracing debug parameters.



Note: Do not change any of these parameters until you have consulted with Dialogic Technical Support. See ***Getting Technical Support on page xxxvi***.

Module Tab Settings

Click the module name in the left panel tree view to modify information about each hardware module.

The right panel Global Settings tab appears.

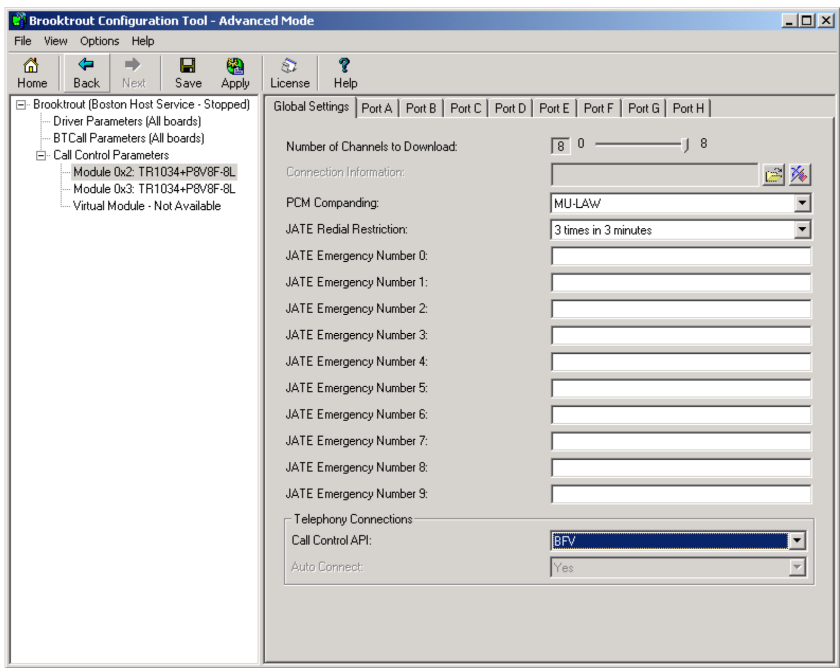
Global Settings Tab

Note: If a virtual module (SR140) is detected, the right panel tabs are different, see ***SR140 Modules on page 126***.

JATE Redial Parameters

If you set the ***country_code*** parameter in the ***btcall.cfg*** file to Japan, the JATE emergency and redial parameters are enabled on the Global Settings tab.

When you set the ***country_code*** parameter to any other value, the JATE parameters are disabled.



Choosing Call Control Type for Call Control Modules

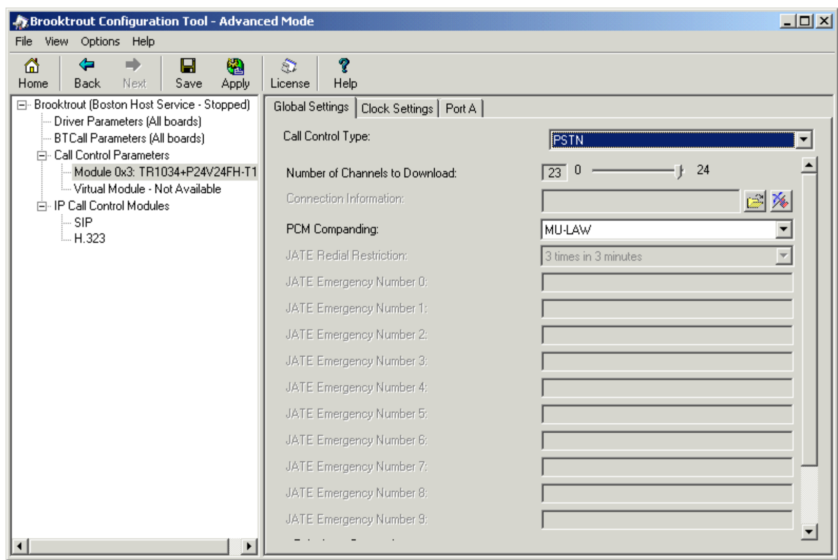
When you choose a module that has Ethernet ports, the Call Control Type parameter appears in the Global Settings tab.

You can configure a module to use PSTN call control or IP call control by setting the Call Control Type parameter.

Note: When using the Configuration Tool, do not manually modify the call control type and model parameters in the ***callctrl.cfg*** file. These are internal configuration tool parameters.

PSTN Call Control Type

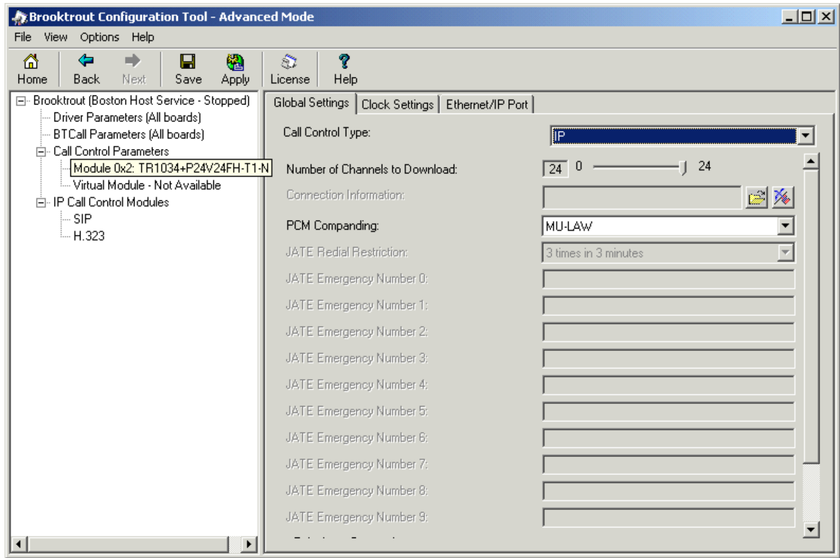
The Global Settings, Clock Settings, and Port A tabs are visible when you select the PSTN option:



Note: Modules that do not have Ethernet ports, do not have the Call Control Type parameter appear in the Global Settings tab.

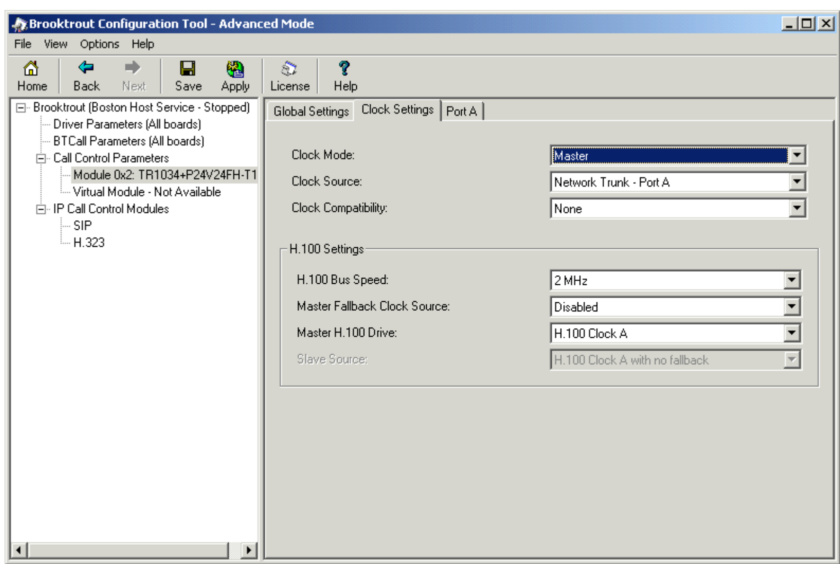
IP Call Control Type

Choose the IP option to associate this Call Control Type with the Dialogic® Brooktrout® module. The configuration tool removes the Port tabs and displays the Ethernet/IP Port tab:



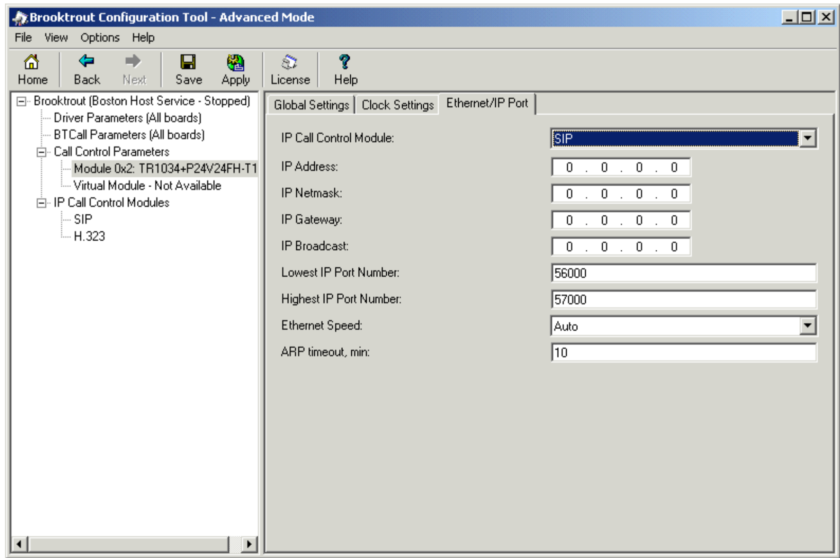
Clock Settings Tab

Click on the Clock Settings tab to view or edit the module clock settings.



Ethernet/IP Port Tab

Only modules equipped with Ethernet ports that have the Call Control Type parameter in the Global Settings tab set to IP show the Ethernet/IP Port tab. By default, the IP Call Control Module parameter is set to the first available IP Call Control Module:

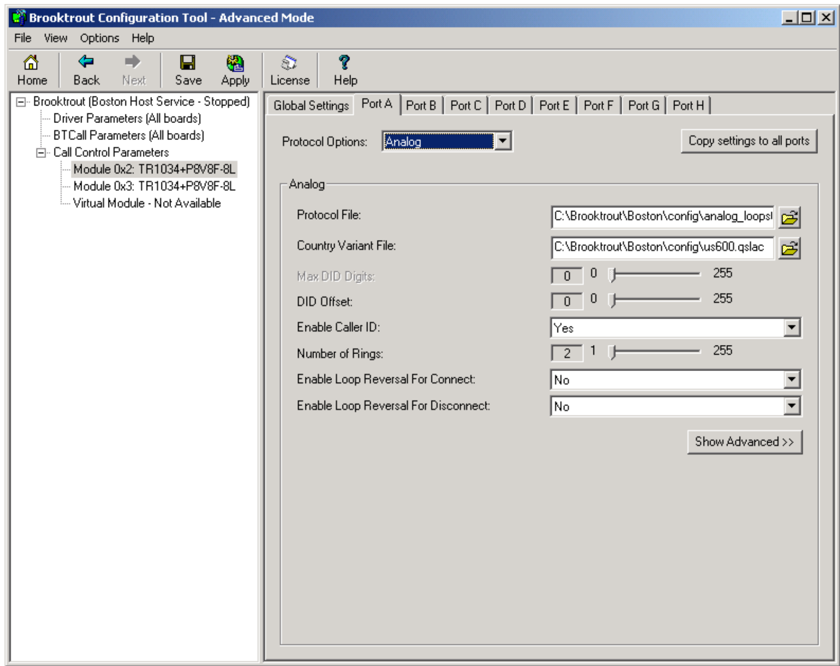


Port *X* Tab

Click on one of the Port tabs to view or edit the port settings.

Choose one of the Port tabs. The Port (*X*) tab appears.

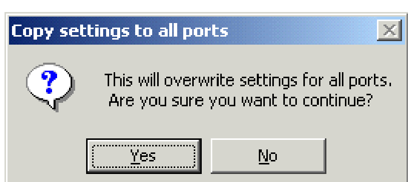
For example, when Port A of Module 0x2 is selected, it shows the Analog port settings.



Copy Settings to All Ports

If this module has more than one port, you can copy these settings when you click on the Copy settings to all ports button.

A Copy settings to all ports dialog box appears.



To confirm your changes, click Yes.

Number of Rings

The Number of Rings parameter (for analog protocol options) specifies the number of rings that the system detects before a system reports a new incoming call to the application.

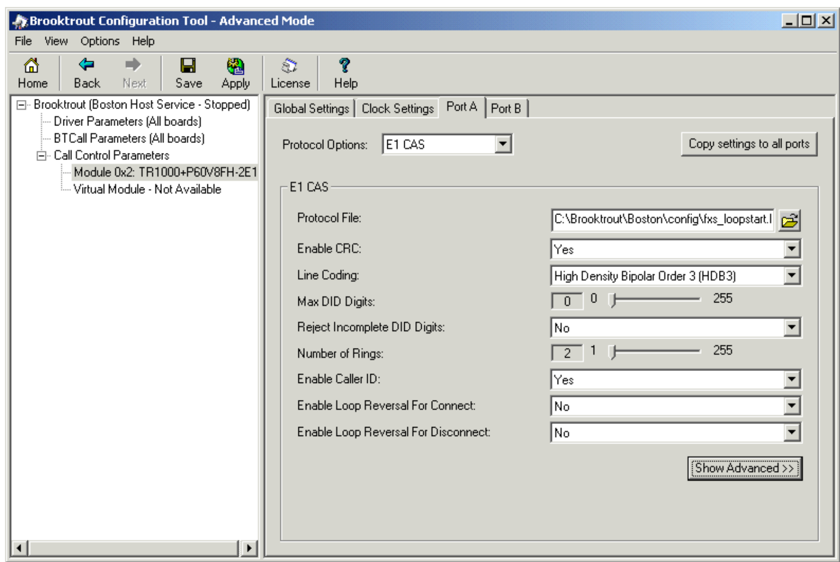
The Number of Rings default is set to 1. However, when you enable the Caller ID parameter, the Number of Rings is set to 2. This has no effect on the minimum value of the Number of Rings parameter.

Note: In North America and some other locations, the system sends the caller ID signal between the first and second rings. To detect the caller ID correctly, you must set the Number of Rings parameter to a value of 2 or greater to prevent the system from reporting the call to the application before the caller ID is sent by the Central Office.

For more information about the Show Advanced >> button, see ***Advanced Settings on page 143***.

E1 CAS Port

When you choose the E1 CAS Protocol option, the following E1 CAS parameters appear:



When using E1 or BRI lines, the phone number is longer than 20 digits. You can include up to 255 digits for Euro_ISDN phone numbers using the Max DID Digits parameter.

SR140 Modules

➢ **When you choose a SR140 module in the left panel, the right panel tabs change.**

SR140 modules features and number of channels depends on the license installed on the user system. You can obtain, install, and update the license using the Dialogic® Brooktrout® License Manager utility.

Note: A special demo license is available in either a single- or two-channel configuration that will run for a 30 day period. Contact your Dialogic sales representative to obtain this license.

The Apply button on the toolbar starts the Boston Host Service to initialize the board. See ***Advanced Mode Buttons on page 113***.

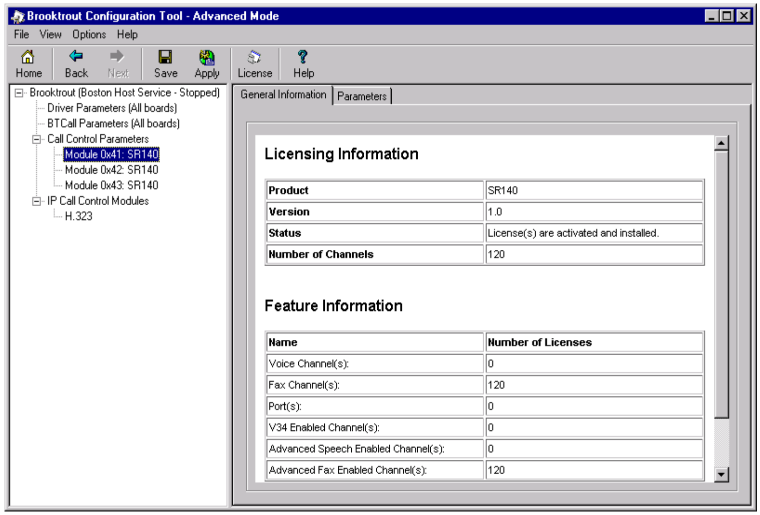
The License button on the toolbar launches the Dialogic® Brooktrout® License Manager (***bkrtlicmgr.exe***).

For more information, see:

- ***Using the Boston Host Service on page 77***
- Activating Dialogic® Brooktrout® Products in the ***Dialogic® Brooktrout® SR140 User Guide***.
- ***Volume 6 of the Bfv API Reference Manual***

General Information Tab for SR140

The Dialogic® Brooktrout® licensing library is queried for the licensing status, number of channels, and features.

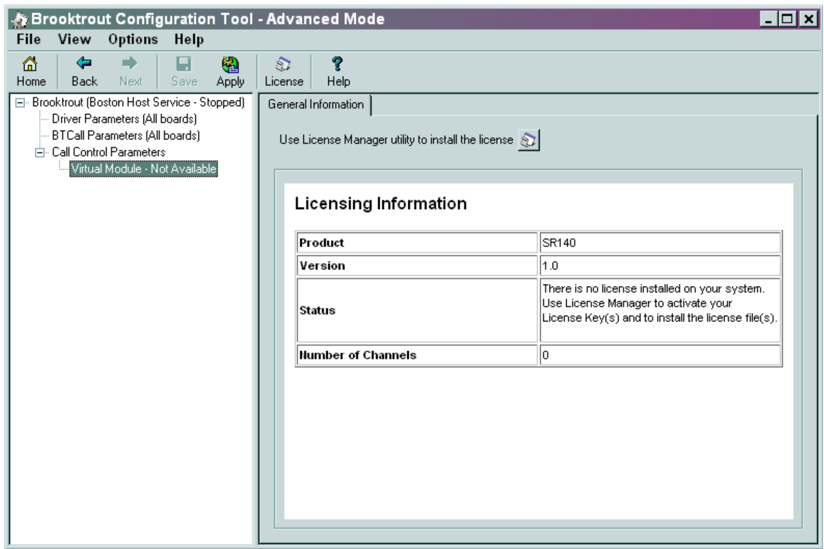

Brooktrout Configuration Tool - Advanced Mode
File View Options Help
Home Back Next Save Apply License Help
Brooktrout (Boston Host Service - Stopped)
Driver Parameters (All boards)
BTCall Parameters (All boards)
Call Control Parameters
Module 0x41: SR140
Module 0x42: SR140
Module 0x43: SR140
IP Call Control Modules
H.323
General Information
Parameters
Licensing Information
Product
SR140
Version
1.0
Status
License(s) are activated and installed.
Number of Channels
120
Feature Information
Name
Number of Licenses
Voice Channel(s):
0
Fax Channel(s):
120
Port(s):
0
V34 Enabled Channel(s):
0
Advanced Speech Enabled Channel(s):
0
Advanced Fax Enabled Channel(s):
120

No License Available

A SR140 virtual module displays in the General Information tab even when there is no valid license installed.

You can click the button after Use License Manager utility to install the license to invoke the Dialogic® Brooktrout® License Manager utility (***brktlicmgr.exe***).

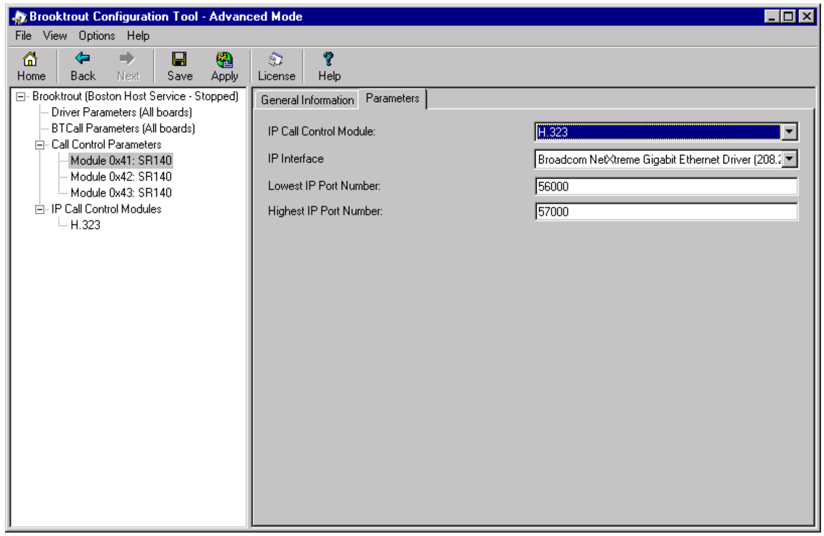
For more information, see Activating Dialogic® Brooktrout® Products in the ***Dialogic® Brooktrout® SR140 User Guide*.**



Parameters Tab for SR140

The Parameters tab displays the call control module type, IP Interface and IP port number information.

Note: The IP Interface parameter applies only to SR140 modules.



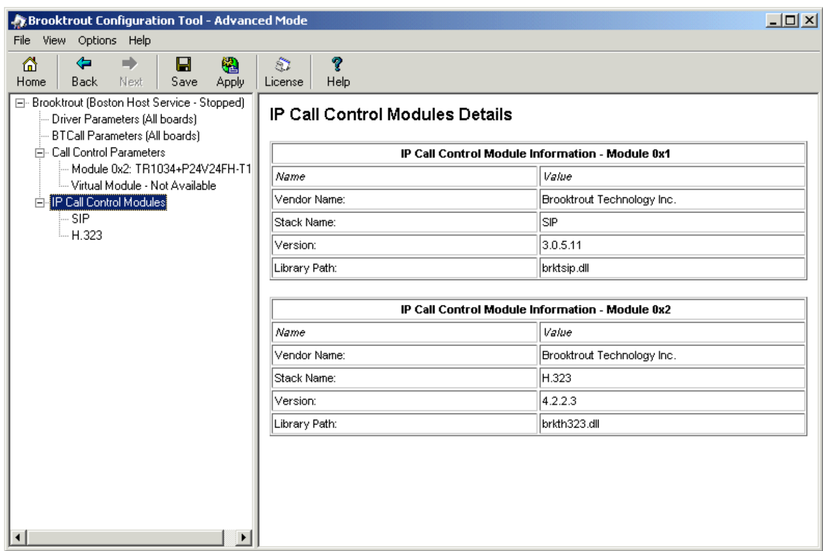
IP Call Control Modules

➢ **Click IP Call Control Modules on the left panel to display information about any IP-enabled modules on the system in the right panel.**

This option is present only when the system has at least one IP-enabled board. Parameters are saved to the ***callctrl.cfg*** file.

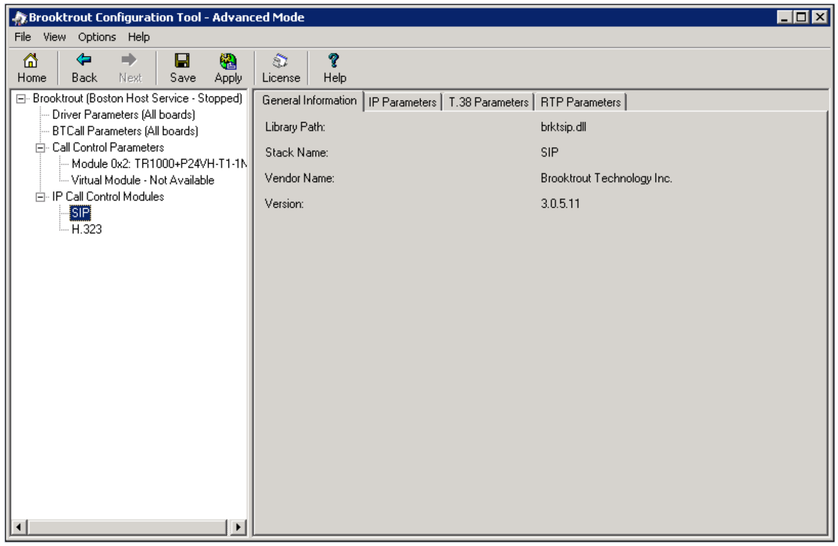
See ***Using a Call Control Configuration File on page 145*** and ***Volume 6*** of the ***Bfv API Reference Manual*** for parameter information.

IP Call Control Modules Details



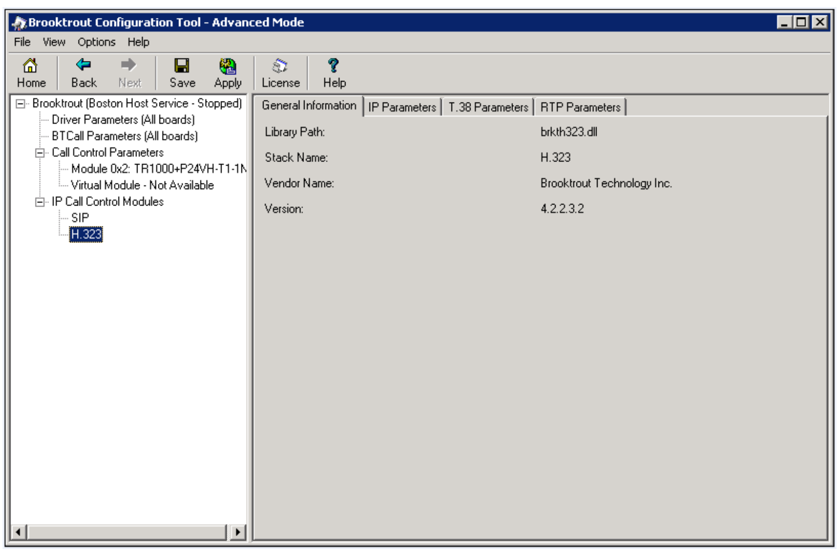
General Information Tab for SIP Stack

The General Information tab contains information about the library path, stack name, vendor name, and version number.



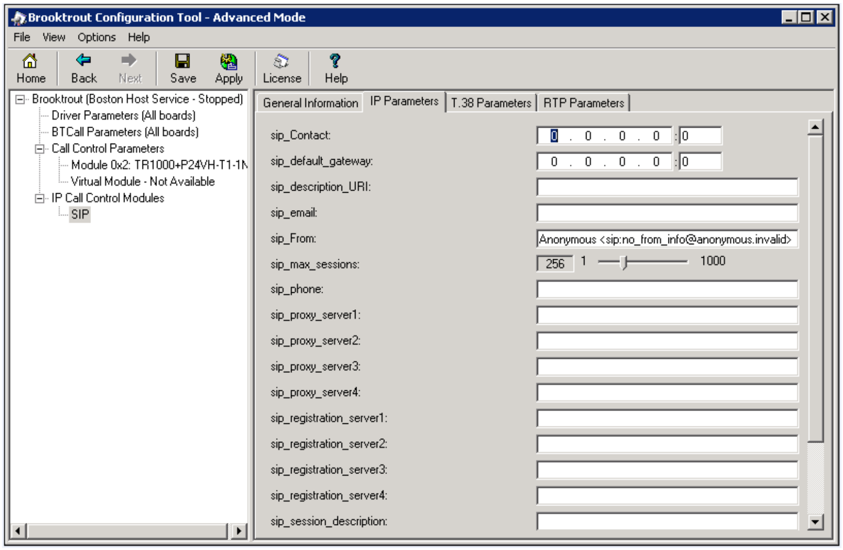
General Information Tab for H.323 Stack

When you select to configure an H.323 stack, the General Information tab details can look like this example:



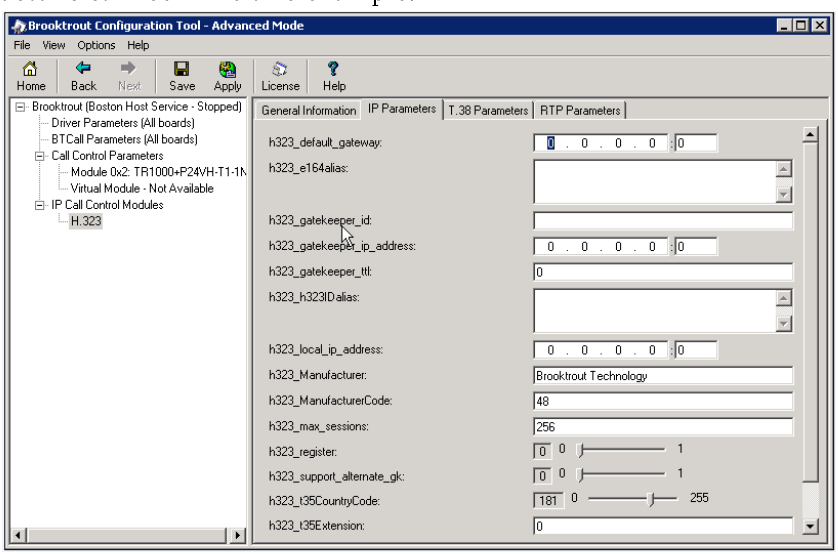
IP Parameters Tab for SIP Stack

The SIP Parameters tab contains the following parameters.



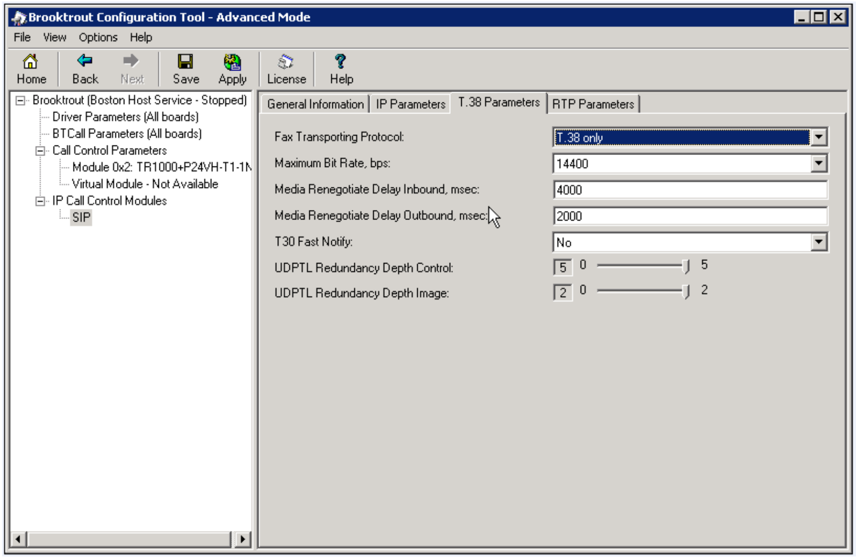
IP Parameters Tab for H.323 Stack

When you select to configure an H.323 stack, the IP Parameters tab details can look like this example:



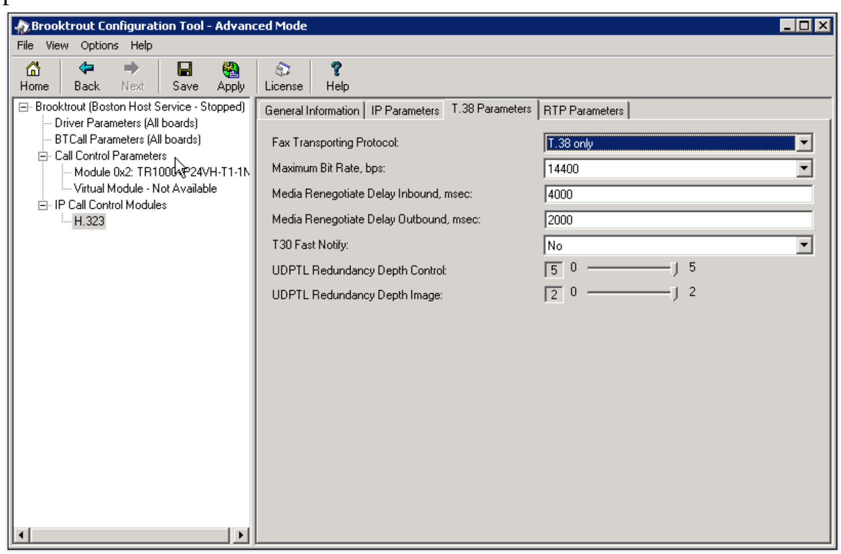
T.38 Parameters Tab for SIP Stack

TheT.38 Parameters tab for a SIP stack contains the following parameters.



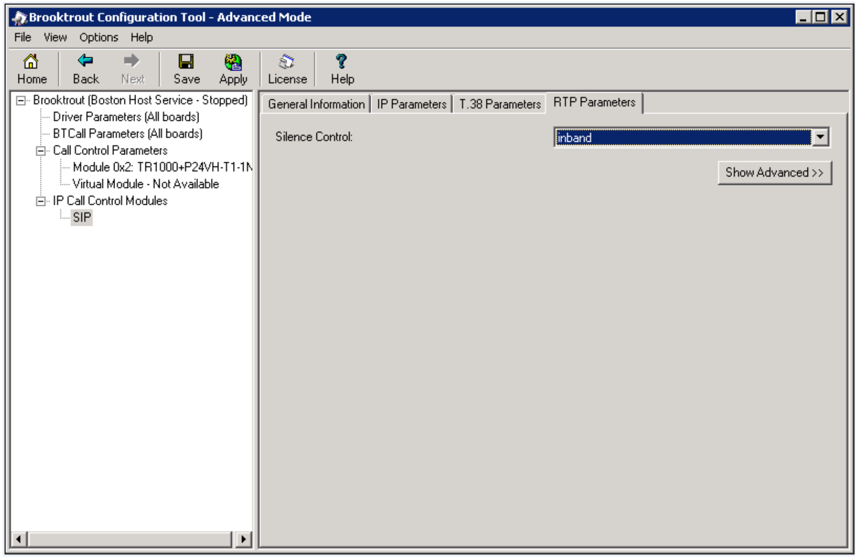
T.38 Parameters for H.323 Stack

The T.38 Parameters tab for an H.323 stack contains the following parameters.



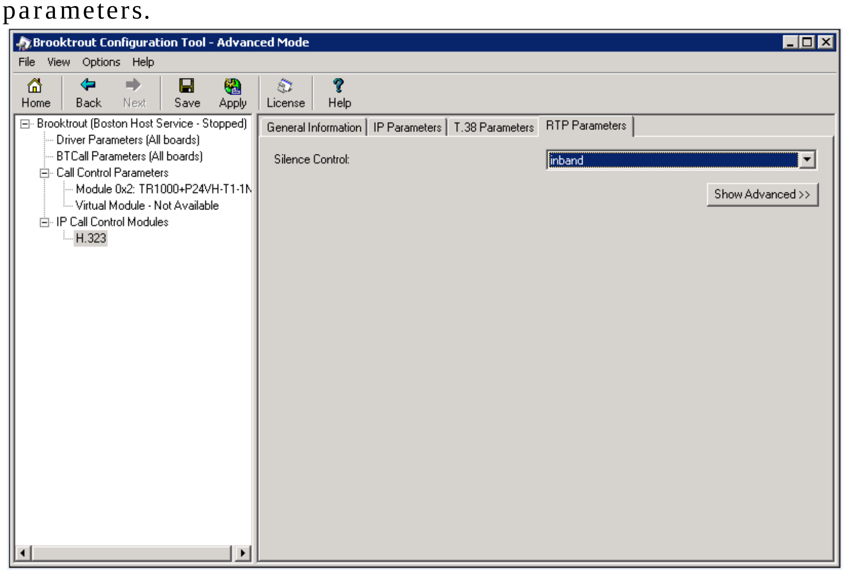
RTP Parameters Tab for SIP Stack

The RTP Parameters tab for a SIP stack contains the following parameters.



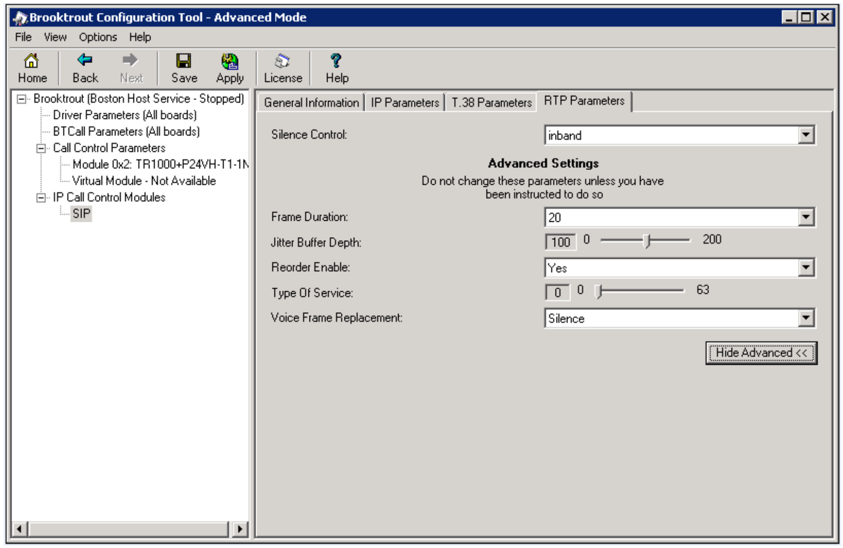
RTP Parameters for H.323 Stack

The RTP Parameters tab for an H.323 stack contains the following parameters.



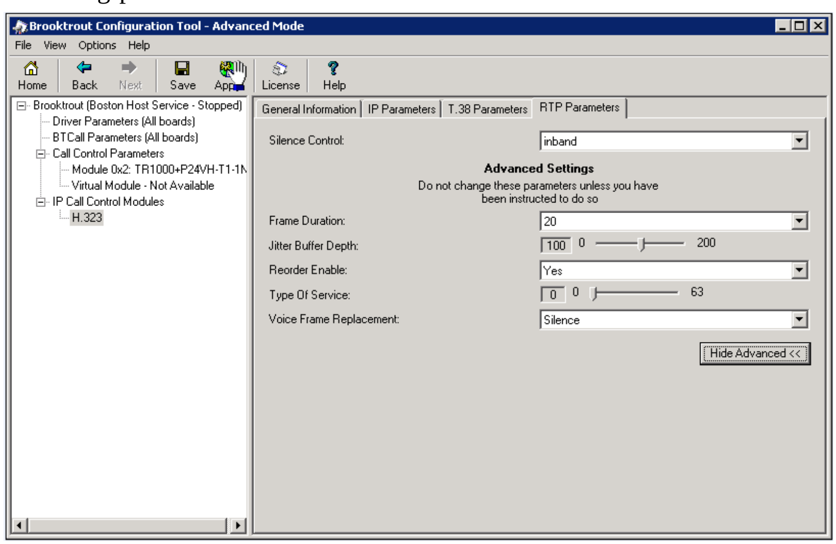
Advanced RTP Parameters Tab for SIP Stack

The advanced RTP Parameters tab for a SIP stack contains the following parameters.



Advanced RTP Parameters for H.323 Stack

The advanced RTP Parameters tab for an H.323 stack contains the following parameters.



(Page left blank intentionally.)

Offline Mode

With Offline Mode you can:

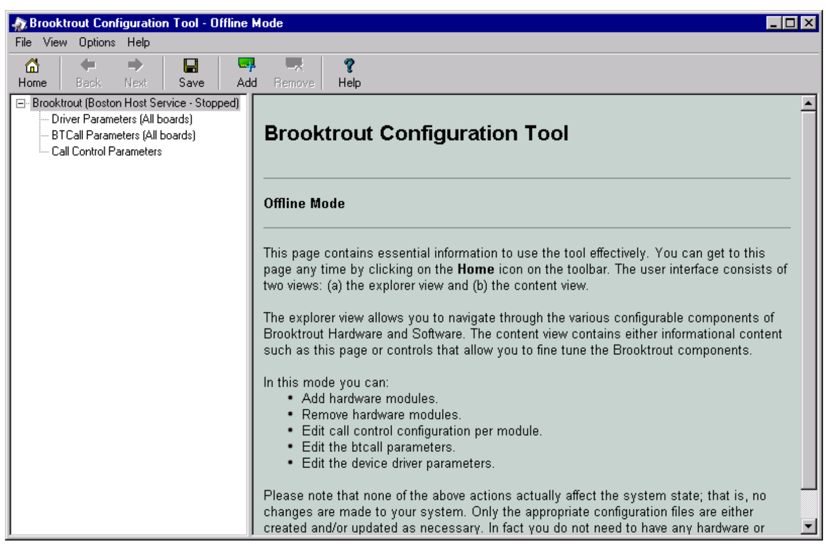
- Create or edit configuration files such as ***btcall.cfg*** and ***callctrl.cfg***.
- Edit the device driver parameters.
- Add or remove, modules (boards).

Offline Mode allows you to create/edit configuration files when no boards are installed.

Offline Mode Window

When you enter Offline Mode, the Brooktrout Configuration Tool - Offline Mode window appears.

To navigate in Offline Mode, use the menus and buttons on the tool bar and the left panel tree view. The buttons are similar to Advanced Mode (see ***Advanced Mode on page 110***) except that the Apply button is not present and Add and Remove buttons are available. These buttons allow you to add/remove items to and from the configuration files or add/remove modules from the configuration.

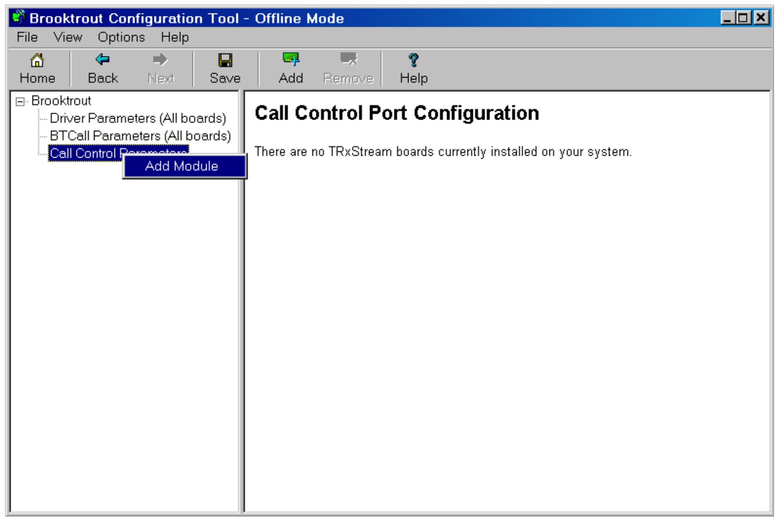


Adding a Module

➢ **Use the right mouse button to add and remove modules.**

1. Right-click on Call Control Parameters to add a module.

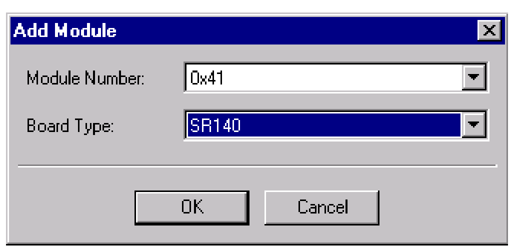
   The following right panel information appears when no modules are installed.



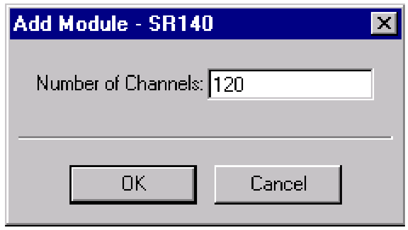
2. The Add Module dialog box appears.

   Select the Module Number and Board Type, then click OK. See *Determining the Board Module Number on page 2* for more information

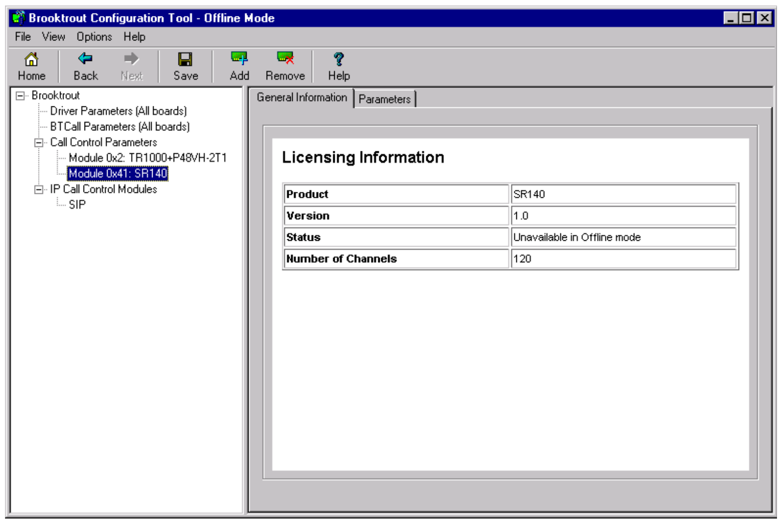
Note: Always set the module number for the SR140 to start with 0x41, because the Configuration Tool Advanced Mode initializes the SR140 module as that default.



3. Enter the correct Number of Channels when a SR140 module type is selected, then click OK:



4. Click the General Information tab to see the added information. This example shows information for a SR140 module:

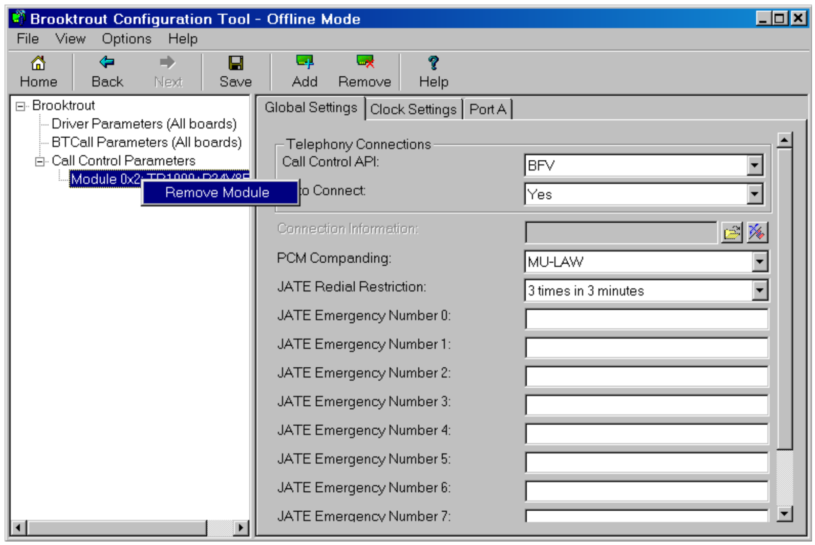


Note: For SR140 modules, you can edit the Number of Channels parameter entry of the call control configuration file.

Removing a Module

➢ **Right-click on a module name to remove it from the configuration.**

In the left panel, select the Remove Module option to remove the module from the configuration.

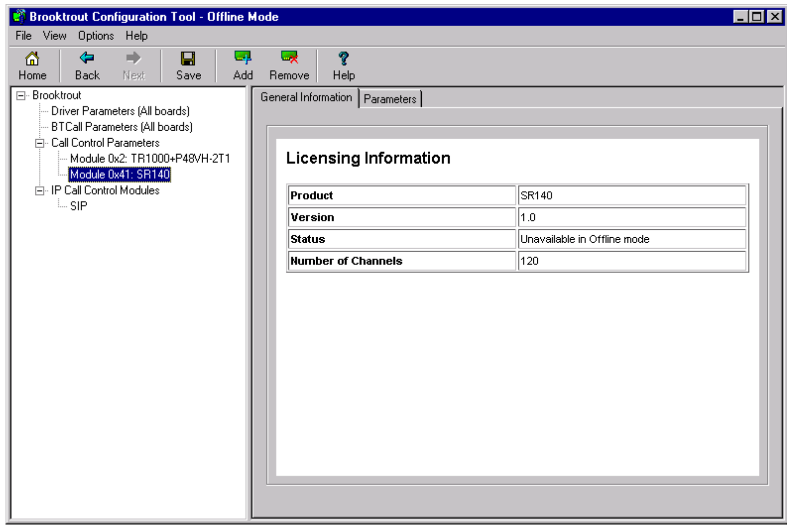


SR140 Modules in Offline Mode

When a SR140 module is detected in Offline Mode, the feature information is unavailable.

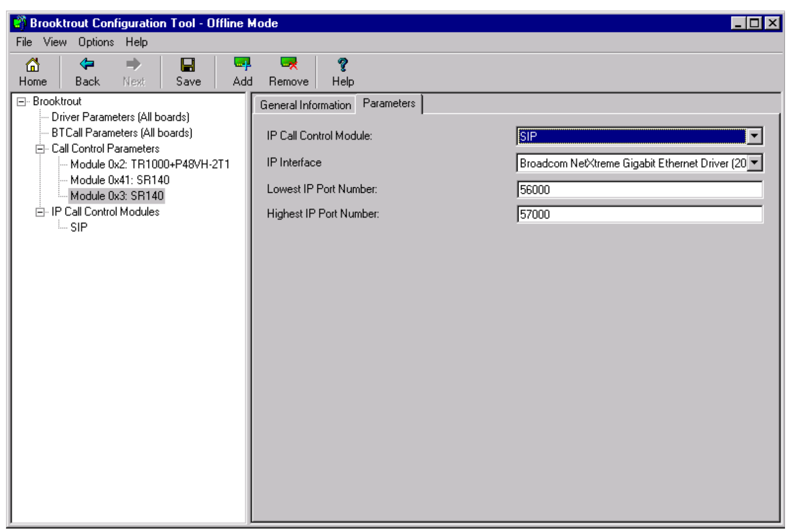
General Information Tab for SR140

The following information is displayed in the General Information panel for SR140 modules:



Parameters Tab for SR140

The Parameters tab displays the editable parameters for the SR140 module.



Silent Mode

To specify Silent Mode, use command line option `/s`, for example:

```
c:\Brooktrout\Boston\utils\winnt\bin\configtool.exe /s
```

You have no interaction once you start the configuration tool. However, you can find information on the actions that were carried out during Silent Mode by referring to the ***configuration.log*** file that is located in the current working directory.

To manually configure the configuration file locations without using the Preferences dialog box (see ***Preferences Dialog Box on page 96***) in Silent Mode, use the command line `/q` option, for example:

```
c:\Brooktrout\Boston\utils\winnt\bin\configtool.exe /q
```

When SR140 modules are present in Silent Mode, this mode does not display any windows or dialog boxes; however, the following occurs:

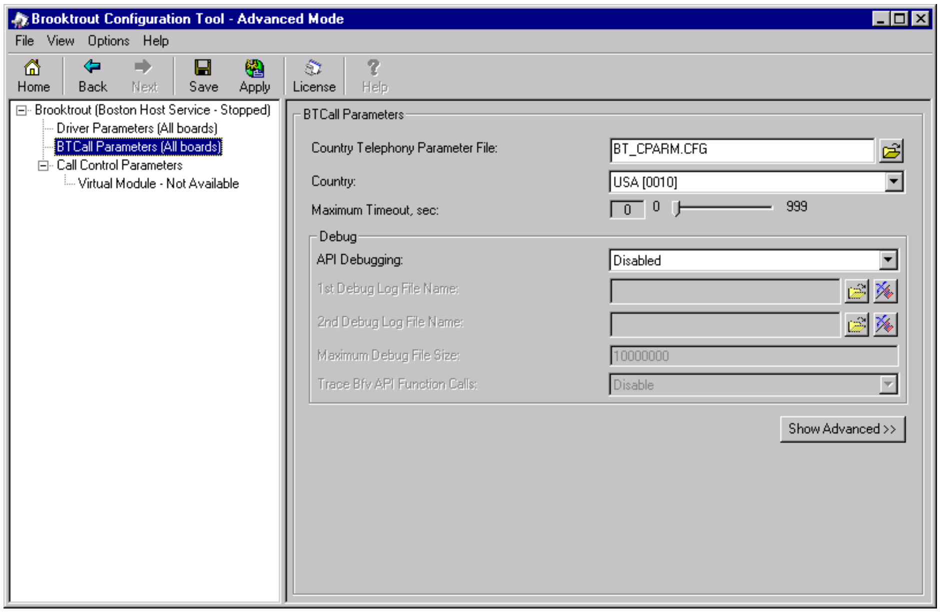
- Reads the configuration file
- Boston device driver starts, if stopped
- Validates data
- Writes information to the configuration file
- Boston Host Service starts

See ***Volume 6*** in the ***Bfv API Reference Manual*** for information about the configuration files.

Advanced Settings

Show Advanced Button

When a parameter panel (for example: Driver Parameters, BTCall Parameters, Call Control Parameters) has advanced settings, the Show Advanced >> button appears on the bottom right of the right panel:

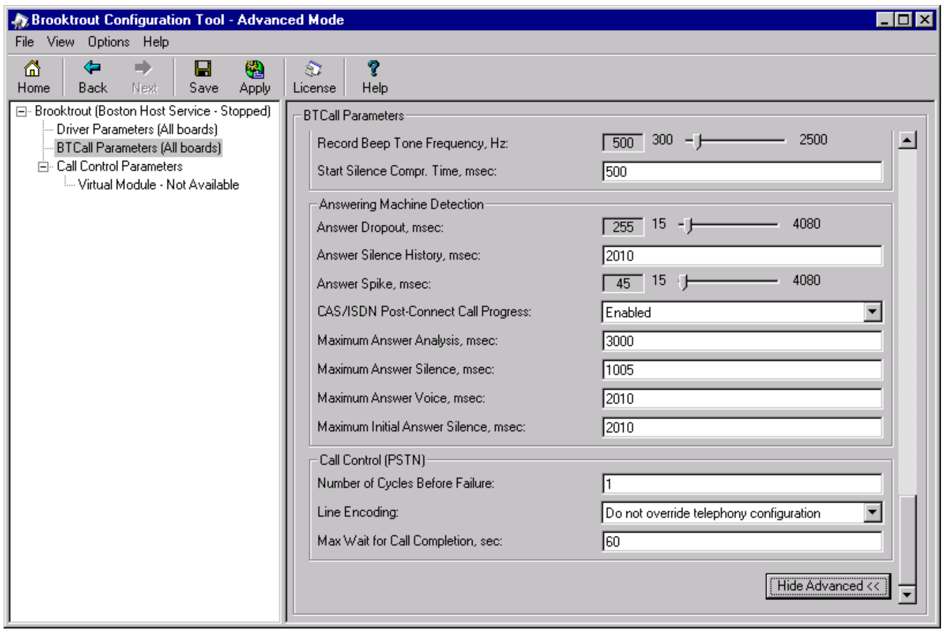


Hide Advanced Button

When you click Show Advanced, the Advanced Settings portion of the panel appears, and the Hide Advanced << button is displayed.

There can be more than one panel of parameter information for Advanced Settings.

Note: Most users never need to alter the Advanced Settings parameter values. If you know these settings are incorrect, please contact Technical Support for more information. See *Getting Technical Support on page xxxvi*.



Using a Call Control Configuration File

The call control configuration file (***callctrl.cfg***) contains parameters that define how to configure your Dialogic® Brooktrout® boards for call control. The ***callctrl.cfg*** file replaces the ***teleph.cfg*** and ***ecc.cfg*** files previously used to define the call control configuration.

If you want to configure call control on your Dialogic® Brooktrout® boards, you can create a ***callctrl.cfg*** file on all supported operating systems (Linux, Solaris, and Windows). On Windows® operating systems, you have the option of using the Dialogic® Brooktrout® Configuration Tool to configure the call control, or you can create or edit a call control configuration file that is read by the configuration tool. See ***Using the Dialogic® Brooktrout® Configuration Tool on Windows Systems on page 91*** and ***Volume 6, Appendix A*** of the ***Bfv API Reference Manual***.

About the Call Control Configuration File

The call control configuration file is an ASCII file that contains general configuration parameters for all telephony hardware modules and static telephony connections to be formed for all modules. You can create your own file or edit one of the sample files supplied with the SDK in the ***Brooktrout\Boston\config\samples.cfg*** directory.

A complete description of the parameters in the call control configuration file along with examples of analog LS, analog DID, BRI, E1 ISDN (PRI), E1 CAS, E1 R2 CAS, T1 ISDN, T1 RBS, and IP call control configuration files is provided in ***Volume 6, Appendix A*** of the ***Bfv API Reference Manual.***

The ***call_control*** parameter in the user-defined configuration (***btcall.cfg***) file specifies the path and filename of the call control configuration file. The Bfv API and the Configuration Tool use ***callctrl.cfg*** as the default value. The application passes the ***callctrl.cfg*** file containing all the telephony configuration information for the BOSTON modules to the ***BfvLineReset*** function. ***BfvLineReset*** uses this information to reset and initialize the system.

The call control configuration (***callctrl.cfg)*** file is divided into the following sections:

- Global parameters for call control that select trace options such as the name and size of the trace file and whether to turn tracing on or off.
- Global module parameters that define connections, specify A-law or µ-law, and specify ***autoconnect*** for the entire module.
- Generic clock configuration parameters that specify clock mode and source and H.100-specific parameters for the entire module.
- Port configuration parameters that describes the signaling protocol of a port and the specific parameters for the port based on the signaling protocol.
- IP call control configuration parameters that describe IP-enabled boards on the system.
- Ethernet interface parameters where you can configure modules with multiple interfaces.

Call Control (callctrl.cfg) File Examples

See ***Volume 6, Appendix A***, in the ***Bfv API Reference Manual*** for configuration file (***callctrl.cfg***) examples, including (but not limited to) the following port-specific configurations:

- Analog ports (loop start (LS) and direct inward dialing DID))
- ISDN BRI ports
- E1 CAS, E1 R2 CAS, and E1 ISDN ports
- T1 ISDN and T1 RBS ports
- SIP ports
- H.323 ports
- T1/E1 QSIG ports

7 - Directory Structure

This chapter contains tables that show the directory structure used for all installed binary, firmware, and sample source files, both platform-independent and platform-specific for all operating system platforms.

Some facilities might not be used with your SDK.

For information on how to compile sample applications using Windows® Developer Studio® Project files, Dialogic® Brooktrout® files, and makefiles, refer to the ***Dialogic® Brooktrout® TR1000 Series SDK Developer Guide.***

Installed Files Directory Structures

The directory structures created when you install the Bfv API are dependent on the operating system platform that you are using:

- *Table 4 - Platform-Independent Files on page 148*
- *Table 5 - Linux-Specific Files on page 150*
- *Table 6 - Solaris-Specific Files on page 152*
- *Table 7 - Windows-Specific Files on page 154*

See one of the following for the default directory:

- Windows: `\Brooktrout\Boston`
- Unix: `/usr/sys/brooktrout/boston`

Table 4. Platform-Independent Files

Directory	Contents
acc.api	Sub-directory for Audio Conferencing (ACC) files
acc.api/inc	ACC library header file
acc.api/samples	Directories for ACC sample files
acc.api/samples/bridgeconf	Sample application for a bridge conference
acc.api/samples/simpleconf	Sample application for a simple conference
bfv.api	Sub-directory for the Bfv API files
bfv.api/app.src	Configuration files and sample application source files
bfv.api/app.src/ivr_msg	Specific infopkt files for the IVR sample application
bfv.api/app.src/params	Country parameter file
bfv.api/bapp.src	Configuration files and BOSTON-specific sample application source files
bfv.api/fonts	Fonts for ASCII fax transmission
bfv.api/inc	Bfv API library header files
bfv.api/misc	High-level sample files

Table 4. Platform-Independent Files (Continued)

Directory	Contents
bfv.api/samples	Low-level sample files
bfv.api/utils/fonts	Font files
bin	Runtime executable, libraries, and help files for audio conferencing, IP call control and Dialogic® Brooktrout® License Manager
bsmi.api	Sub-directory for BOSTON Simple Message Interface files
bsmi.api/bapp.src	BSMI sample source files
bsmi.api/inc	BSMI library header files including country-specific R2 CAS protocol files
config	Sub-directory for call control configuration files and protocol variants
config/samples.cfg	Sample configuration files
driver	Driver directories
driver/common	Files used by low level applications to communicate with driver
driver/inc	Header files used by both driver and low level applications
driver/fw/cmndset	Firmware command set header files
fw	Firmware files
utils/oslib	Sub-directory for operating system independent files
utils/osilib/inc	Operating system independent library header file
utils/oslib/osi.src	Operating system independent source files

Table 5. Linux-Specific Files

Directory	Contents
acc.api/linux	Sub-directory for Linux-specific Audio Conferencing (ACC)
acc.api/linux/lib/3.4	Compiled library files for ACC for Enterprise Linux ES/AS 4.0
acc.api/linux/lib/7	Compiled library files for ACC for Enterprise Linux ES/AS 3.0 or earlier
acc.api/linux/lib/4.1	Compiled library files for ACC for Enterprise Linux Edition 5.0
acc.api/linux/samples	Sub-directory for sample ACC applications
acc.api/linux/samples/bridgeconf	Makefile, configuration files, and program files for a bridge conference
acc.api/linux/samples/simpleconf	Makefile, configuration files, and program files for a simple conference
bfv.api/linux	Sub-directory for Linux-specific compilation areas and utilities for Bfv API
bfv.api/linux/app.src	Makefile and configuration files for sample applications
bfv.api/linux/bapp.src	Makefile, configuration files, and compiled executables for BOSTON-specific sample applications
bfv.api/linux/bin/3.4	Bfv API binary files for Enterprise Linux ES/AS 4.0
bfv.api/linux/bin/7	Bfv API binary files for Enterprise Linux ES/AS 3.0 or earlier
bfv.api/linux/bin/4.1	Bfv API binary files for Enterprise Linux 5.0
bfv.api/linux/lib/3.4	Compiled Bfv API libraries for Enterprise Linux ES/AS 4.0
bfv.api/linux/lib/7	Compiled Bfv API libraries for Enterprise Linux ES/AS 3.0 and earlier
bfv.api/linux/lib/4.1	Compiled Bfv API libraries for Enterprise Linux 5.0
bfv.api/linux/utils/bin	Compiled G3 utilities

Table 5. Linux-Specific Files (Continued)

Directory	Contents
bsmi.api/linux	Sub-directory for Linux-specific BOSTON Simple Message Interface files
bsmi.api/linux/bapp.src	Makefile
bsmi.api/linux/lib/3.4	Compiled library files for BSMI for Enterprise Linux ES/AS 4.0
bsmi.api/linux/lib/7	Compiled library files for BSMI for Enterprise Linux ES/AS 3.0 and earlier
bsmi.api/linux/lib/4.1	Compiled library files for BSMI for Enterprise Linux 5.0
driver/linux	Sub-directory for Linux-specific driver compilation areas and utilities
driver/linux/common/3.4	Enterprise Linux ES/AS 4.0 driver libraries
driver/linux/common/7	Enterprise Linux ES/AS 3.0 (and earlier) driver libraries
driver/linux/common/4.1	Enterprise ES/AS 5.0 driver libraries
driver/linux/install	Installation program and library
driver/linux/kernel	Compiled binaries for the driver
driver/linux/user	Compiled binaries for the driver utility programs
utils/linux	Sub-directory for Linux-specific tracing utilities and configuration files
utils/osilib/linux/lib	Sub-directory for compiled OSI libraries for Linux
utils/osilib/linux/lib/3.4	Compiled OSI libraries for Enterprise Linux ES/AS 4.0
utils/osilib/linux/lib/7	Compiled OSI libraries for Enterprise Linux ES/AS 3.0 and earlier
utils/osilib/linux/lib/4.1	Compiled OSI libraries for Enterprise Linux 5.0
utils/osilib/linux/osi.src	Makefiles for OSI libraries

Table 6. Solaris-Specific Files

Directory	Contents
acc.api/solaris	Sub-directory for Solaris-specific Audio Conferencing (ACC) files
acc.api/solaris/lib/32	Compiled library files for ACC for Solaris 32-bit systems
acc.api/solaris/lib/64	Compiled library files for ACC for Solaris 64-bit systems (SPARC only)
acc.api/solaris/samples	Sub-directories for sample ACC applications
acc.api/solaris/samples/bridgeconf	Makefile, configuration files, and program files for a bridge conference
acc.api/solaris/samples/simpleconf	Makefile, configuration files, and program files for a simple conference
bfv.api/solaris	Sub-directory for Solaris-specific compilation areas, and utilities for Bfv API
bfv.api/solaris/app.src	Makefile and configuration files
bfv.api/solaris/bapp.src	Makefile, configuration files, and compiled executables for BOSTON-specific sample applications
bfv.api/solaris/bin/32	Bfv API binary files for Solaris 32-bit systems
bfv.api/solaris/lib/32	Compiled Bfv API libraries for 32-bit systems
bfv.api/solaris/lib/64	Compiled Bfv API libraries for 64-bit systems (SPARC only)
bfv.api/solaris/utils	Sub-directory for G3 utilities
bfv.api/solaris/utils/bin	Compiled G3 utilities
bsmi.api/solaris	Sub-directories for Solaris-specific BOSTON Simple Message Interface files
bsmi.api/solaris/bapp.src	Makefile
bsmi.api/solaris/lib/32	Compiled library files for BSMI for Solaris 32-bit systems
bsmi.api/solaris/lib/64	Compiled library files for BSMI for Solaris 64-bit systems (SPARC only)

Table 6. Solaris-Specific Files (Continued)

Directory	Contents
driver/solaris	Sub-directory for Solaris-specific compilation areas and utilities
driver/solaris/common/32	Solaris-specific 32-bit driver libraries
driver/solaris/common/64	Solaris-specific 64-bit driver libraries
driver/solaris/install	Installation program
driver/solaris/kernel	
driver/solaris/kernel/27	Compiled binaries for the 32-and 64-bit drivers
driver/solaris/user	Sub-directory for compiled binaries for the driver utility programs
utils/osilib/solaris/lib	Sub-directory for compiled OSI libraries for Solaris
utils/osilib/solaris/lib/32	Compiled OSI libraries for Solaris 32-bit systems
utils/osilib/solaris/lib/64	Compiled OSI libraries for Solaris 64-bit systems
utils/osilib/solaris/osi.src	Makefiles for OSI libraries
utils/solaris	Configuration files for Solaris tracing utilities
utils/solaris//32	Tracing utility for 32-bit systems
utils/solaris//64	Tracing utility for 64-bit systems
Solaris SPARC Systems	
usr/lib/sparcv9	Compiled library files for SPARC V9 operating systems

The Call Tracer utility, and a sample configuration file, can be found in the ***\Brooktrout\Boston\utils\winnt\bin*** directory when installing the SDK, or in the ***\Brooktrout\bin*** directory when installing just the Brooktrout System Software.

Table 7. Windows-Specific Files

Directory	Contents
acc.api	Files for Windows-specific Audio Conferencing (ACC)
acc.api\winnt\inc	Library header files
acc.api\winnt\lib	Compiled library files for ACC
acc.api\winnt\samples	Sub-directory for ACC sample applications
acc.api\winnt\samples\bridgeconf	Makefile, configuration files, and program files for a bridge conference
acc.api\winnt\samples\bridgeconf\res	Resource files
acc.api\winnt\samples\simpleconf	Makefile, configuration files, and program files for a simple conference
acc.api\winnt\samples\simpleconf\res	Resource files
bfv.api\winnt	Sub-directory for Windows-specific compilation areas and utilities
bfv.api\winnt\app.src	Makefile, configuration files, and sample application files
bfv.api\winnt\bapp.src	Makefile, configuration files, and compiled executables for BOSTON-specific sample applications
bfv.api\winnt\bin	Boston Host Service executable and library
bfv.api\winnt\lib	Compiled Bfv API libraries
bfv.api\winnt\utils	Sub-directory for G3 utilities
bfv.api\winnt\utils\bin	Compiled G3 utilities
bsmi.api\winnt	Sub-directory for Windows-specific BOSTON Simple Message Interface files
bsmi.api\winnt\lib	Compiled library files for BSMI
bsmi.api\winnt\bapp.src	Makefile

Table 7. Windows-Specific Files (Continued)

Directory	Contents
bsmi.api\winnt\bin	Debugging program and configuration file.
driver\winnt	Sub-directory for Windows® platform-specific compilation areas and utilities
driver\winnt\common	Windows-specific driver libraries
driver\winnt\install	Installation program and associated files for 32- and 64-bit operating systems
driver\winnt\user	Compiled binaries for the driver utility programs
driver\winnt\pnp	Plug-and-Play ***.inf*** file
driver\winnt\pnp\TRxStream	PnP boston driver, driver symbol file, co-installer and device property pages for 32- and 64-bit operating systems.
utils\winnt\bin	Dialogic® Brooktrout® Configuration Tool, AccuCall, and Call Tracer executables, help and configuration files
utils\osilib\winnt\lib	Compiled OSI libraries
utils\osilib\winnt\osi.src	Makefiles for OSI library
%SYSTEMROOT%\system32	Boston Host Service executable and library for 32-bit systems.
%SYSTEMROOT%\syswow64	Boston Host Service executable and library for 64-bit systems.
Note: ***%SYSTEMROOT%*** is usually ***C:\winnt*** or ***C:\windows.***	

8 - Dialogic® Brooktrout® AccuCall and the AccuCall Wizard

This chapter describes how one can use the Dialogic® Brooktrout® AccuCall Wizard to train your board to recognize tones from your PBX and how to create and edit Programmable Call Progress Monitoring (PCPM) tables for use with different Private Branch Exchanges (PBXs).

Note: Dialogic® Brooktrout® AccuCall is used primarily for Dialogic® Brooktrout® Analog TR1000/TR1034 and TruFax® modules.

This chapter contains the following sections:

- ***Dialogic® Brooktrout® AccuCall Wizard on page 159***
- ***Call Progress Tones on page 159***
- ***Using Dialogic® Brooktrout® AccuCall Wizard on page 163***
- ***Custom Tones on page 174***

You can use the Dialogic® Brooktrout® AccuCall Wizard to quickly and easily create a PCPM table that corresponds to a specific PBX, and then use Brooktrout's AccuCall to modify the PCPM table, if needed.

- ***Using Dialogic® Brooktrout® AccuCall on page 206***
- ***Improving Tone Detection on page 219***

Brooktrout AccuCall enables you to create and edit PCPM tables used for PCPM tone recognition. See ***Call Progress Tones on page 159*** for an overview of basic Call Progress Code concepts. Your applications use PCPM tables to recognize tones generated by various PBXs. AccuCall also enables you to control parameters used to filter frequencies and detect tones.

Dialogic® Brooktrout® AccuCall Wizard

Dialogic® Brooktrout® AccuCall Wizard is accessible from the toolbar in the AccuCall utility.

You can use AccuCall Wizard to quickly and easily create a Programmable Call Progress Monitoring (PCPM) table that corresponds to a specific PBX, and then use AccuCall to modify the tone table, if needed.

The section is organized as follows:

- *Call Progress Tones on page 159*
- *Using Dialogic® Brooktrout® AccuCall Wizard on page 163*
- *Custom Tones on page 174*

Call Progress Tones

This section explains call progress tones, and how your board identifies and responds to them.

Using Call Progress Tones

Your PBX uses audible tones called call progress tones to indicate the progress of a call. These tones include the dial tone, busy tone, and ringing tone like those used by the public telephone network. A caller or computer telephony application uses call progress tones to determine how to respond during a call. For example, the busy signal tells a caller or application to hang up and try the call again later.

Different tones vary in frequencies and cadence.

- A ***call progress tone*** usually has one or two frequencies, referred to as single-frequency or dual-frequency tone. Individual frequencies can range between 300 Hz and 3700 Hz.
- A ***cadence*** is a repeating sequence of sounds. Each element (sound period) in the cadence has distinct frequencies. The elements can also be elements of silence with frequencies of zero Hz. Tones can either have a single cycle or a double cycle. A double cycle tone comprises a primary and secondary cadence, as shown in ***Figure 1 on page 160***.

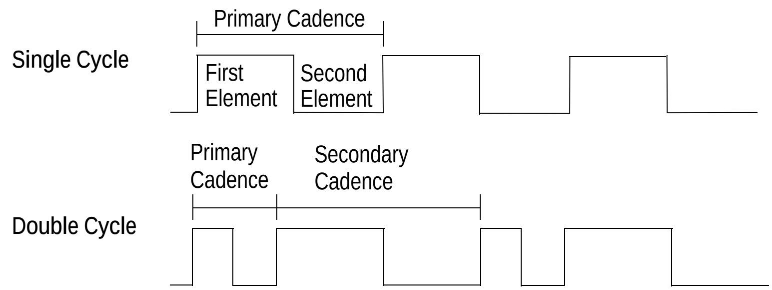


Figure 1. Primary and Secondary Cadences

Table 8 shows some common tones for the U.S. public phone network.

Table 8. Common Tone Frequencies and Cadences

		Primary Cadence	
Tone	**Frequency**	**First Element**	**Second Element**
Trunk Busy	480 Hz and 620 Hz	250 ms	250 ms
Station Busy	480 Hz and 620 Hz	500 ms	500 ms
Ringing	440 Hz and 480 Hz	2 sec.	4 sec.

Recognizing Tones

Since PBXs use different frequencies and cadences for their tones, you need to train your board to recognize specific tones.

The AccuCall Wizard in the AccuCall utility automatically trains your board to recognize the hangup, busy, and ringing tones of your PBX.

Hangup tone

Indicates that the other line has disconnected.

Busy tone

Indicates that the line you called is busy. There are two types of busy signals: trunk busy and station busy. Trunk busy, also called fast busy or reorder tone, indicates that all paths in the network are busy. Station busy means that the number or extension you called is busy.

Ringing tone

Indicates to the called station that there is an incoming call. A simulated ringing signal is returned to the caller to indicate that a path has been established and the called number or extension is being rung. The simulated signal is called ringback.

You can customize the AccuCall Wizard to recognize custom tones such as a dial tone or any other tones used by your PBX with a frequency of 300 Hz to 3700 Hz.

The AccuCall Wizard stores information about each tone in a tone table. For each tone, the tone table contains the frequencies, cadences, tone-specific characteristics, and a Call Progress Code associated with each tone. The AccuCall Wizard automatically assigns a code to the preconfigured tones.

Responding to Incoming Tones

When your board receives a call progress tone, it compares the frequencies and cadences to the tones stored in the tone table. If it finds a match, the driver returns the Call Progress Code associated with the tone. The Call Progress Code conveys information about what type of tone it is and how the application responds. A list of Call Progress Codes are defined in the ***btlib.h*** file.

For example, if your board detects a busy tone, the driver compares the tone to those stored in the tone table. When it matches the incoming tone to the stored busy tone, the driver reports the Call Progress Code (CS_BUSY1) to the application. CS_BUSY1 indicates that the line is busy and that the application needs to try the call again later.

Using Dialogic® Brooktrout® AccuCall Wizard

This section describes how to use the AccuCall Wizard in the AccuCall utility to create a tone table for a specific PBX.

- ***Navigating the Dialogic® Brooktrout® AccuCall Wizard on page 163***
- ***Training a PBX With the Dialogic® Brooktrout® AccuCall Wizard on page 164***
- ***Troubleshooting on page 173***

Navigating the Dialogic® Brooktrout® AccuCall Wizard

All of the AccuCall Wizard windows have four buttons along the bottom:

These navigational buttons function as follows:

Back
: Displays the previous window.

Next
: Displays the following window.

: The Next button is disabled until you have finished all necessary procedures on the current window.

Cancel
: Closes the AccuCall Wizard window.

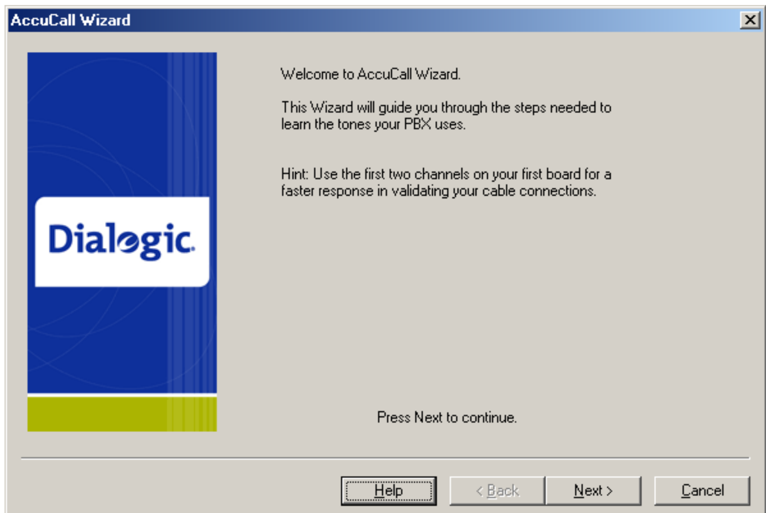
Help
: Displays Help for that window.

Training a PBX With the Dialogic® Brooktrout® AccuCall Wizard

The AccuCall Wizard windows automatically guide you through the process of creating a tone table, so that your board can recognize Call Progress tones from your PBX. This process is called ***training*** the PBX.

AccuCall Wizard Welcome Window

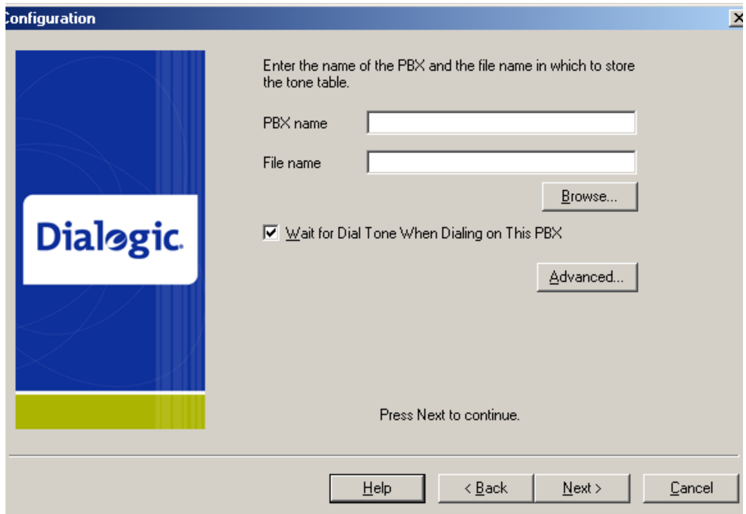
The AccuCall Wizard window appears when you run the AccuCall Wizard.



From the Welcome window you can either cancel the AccuCall Wizard or select Next to display the Configuration dialog box.

Configuration Dialog Box

This Configuration dialog box provides text entry fields where you can enter the name of your PBX and the file name of your tone table.



The AccuCall Wizard automatically inserts the name of the PBX as the file name. To manually set the file name, select the file name and enter the name you want to use. Once you have edited the file name, changing the PBX name does not affect it.

The Configuration dialog box buttons function as follows:

Browse...

Displays the Save As dialog box, where you can choose the directory to save your tone table.

Wait for Dial Tone When Dialing on the PBX

Causes the AccuCall Wizard to wait for a dial tone before making a call. Deselect this option if the dial tone on your PBX is not continuous. Selected by default.

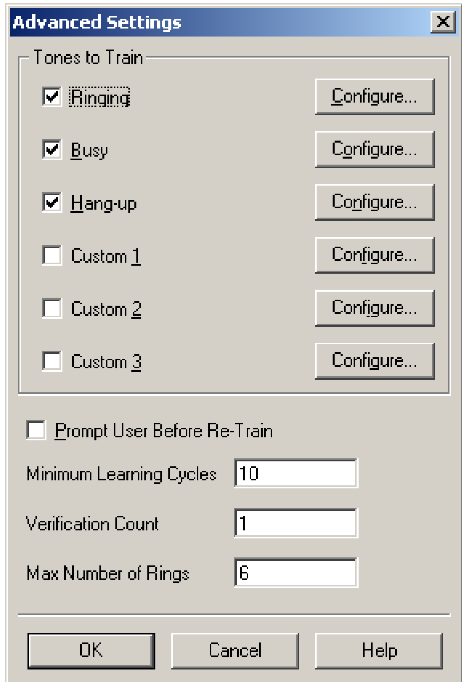
Advanced...

Displays the Advanced Settings dialog box appears when you run the AccuCall Wizard. Select this button to configure custom tones or change the default selections for Ringing, Busy, and Hangup.

When you are finished with the Configuration dialog box, select Next to display the First Extension dialog box.

Advanced Settings Dialog Box

The Advanced Settings dialog box provides the following options:



Ringing	Tells the AccuCall Wizard to train the Ringing tone. Selected by default.
Busy	Tells the AccuCall Wizard to train the Busy signal. Selected by default.
Hangup	Tells the AccuCall Wizard to train the Hang-up signal. Selected by default.
Custom 1/2/3	Tells the AccuCall Wizard to train a custom designed tone. Deselected by default.
Configure	Displays the Custom Tone Configuration dialog box. Deselected by default.

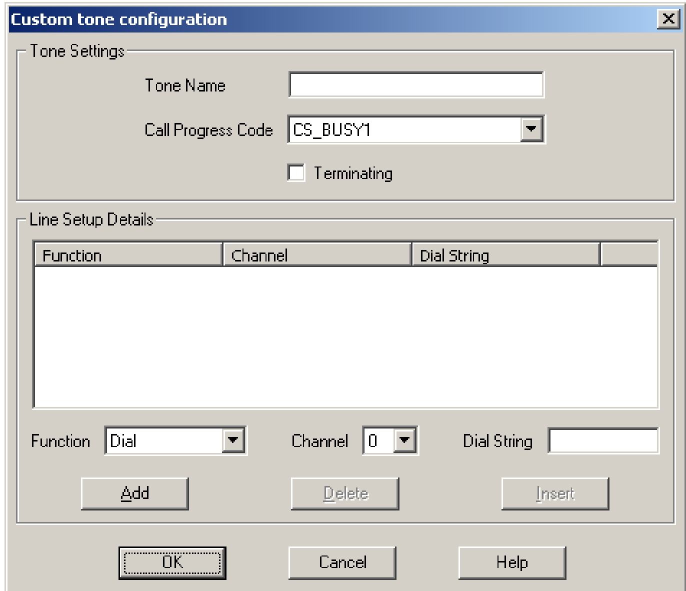
- For Ringing, Busy and Hang-up tones, this dialog box shows how the pre-designed tones were configured.
- For Custom tones, use this dialog box to design your custom tone. After you design a custom tone, the selection box is checked.

Prompt User Before Re-Train	Causes the AccuCall Wizard to display the Test Results dialog box for any failed tones. If there were no problems during testing, the AccuCall Wizard does not display the dialog box.
Minimum Learning Cycles	Defines the number of cadence cycles to gather before completing. Ensure that your PBX provides at least this many cadence cycles. Some PBXs generate a tone only a certain number of times before generating silence or a different type of tone. Default is 10.
Verification Count	Specifies the number of times to verify the tone. Default is 1.
Max Number of Rings	Sets number of rings before the AccuCall Wizard reports no answer.
OK	Returns to the Configuration dialog box.

Note: For users accustomed to previous versions of Dialogic® Brooktrout® AccuCall, ensure that the flash hook duration is now set using the Dialogic® Brooktrout® Configuration Tool. (In the Bfv API, as shown, this value can still be set using AccuCall).

Custom Tone Configuration Dialog Box

Use the Custom Tone Configuration dialog box to designate a custom tone for the AccuCall Wizard to store in the PCPM table.



The Custom Tone Configuration dialog box provides the following options:

Tone Name
Defines the name of the custom tone.

Call Progress Code
Value to report when this tone is detected. Used for call progress analysis. See *Custom Tones on page 174*.

Terminating
Reports during playback/recording (this particular tone). It is up to the application to determine if it wants to terminate play/record. The application typically does this with a user function in the play/record function. (If this particular tone is detected, then set the user function to 1.) If terminating is not selected, then this tone is not reported during play/record.

Function
Selects between Off hook, On hook, Dial, Wait for call, and Pause. Default is Off hook.

Channel
Selects the channel to perform the instruction. Set this option so that channel 0 receives the custom tone. The default is 0. Channel 0 is the first ordinal channel in the system.

Dial String
Provides a text entry box where you can enter a phone number for the AccuCall Wizard to dial.

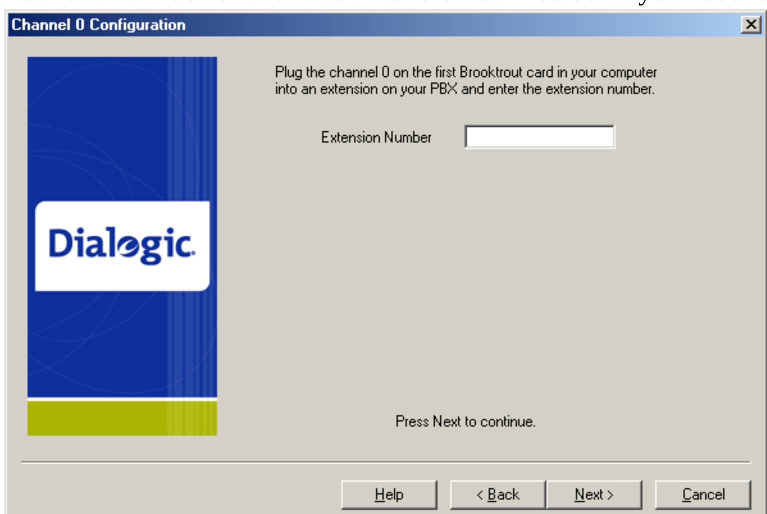
Add	Adds the selected Function, Channel, and Dial String to the list.
Delete	Deletes a selection from the list. This option is available only after you have added an instruction to the list.
Insert	Inserts an instruction to the list above a selected item. This option is available only after you have added an instruction to the list.
OK	Saves the configuration and closes the Custom Tone Configuration dialog box, then re-opens the Advanced Settings dialog box.

Select OK from the Advanced Settings dialog box to go back to the Tone Configuration dialog box.

Select Next to display the First Extension dialog box.-

First Extension Dialog Box

This First Extension dialog box provides a text entry box where you enter the number of the first extension connected to your board.



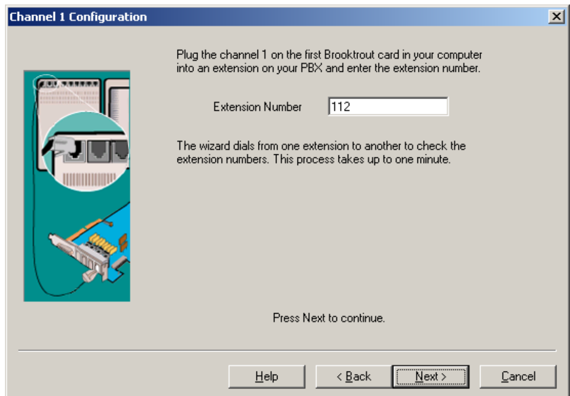
You must provide the AccuCall Wizard with the extensions plugged into the board only if you are training the ringing, busy, or hangup tones. If you have deselected these tones in the Advanced Settings dialog box, the AccuCall Wizard does not display the First Extension dialog box.

When you select Next, the AccuCall Wizard checks to make sure it can detect a phone line at the first extension. If the AccuCall Wizard does not detect a line, an error message appears that says: No loop current detected on first channel. Make sure the line is properly connected. Ensure that the phone line is connected at both ends. You can also plug an ordinary phone into the extension to check if the socket is working.

You cannot proceed to the Second Extension dialog box until the AccuCall Wizard verifies that the first extension is present.

Second Extension Dialog Box

This Second Extension dialog box provides a text entry box where you can enter the number of the second extension connected to your board.



The AccuCall Wizard needs to know the extensions plugged into the board only if you are training the ringing, busy, or hangup tones. The AccuCall Wizard does not display the Second Extension dialog box if you have deselected these tones in the Advanced Settings dialog box.

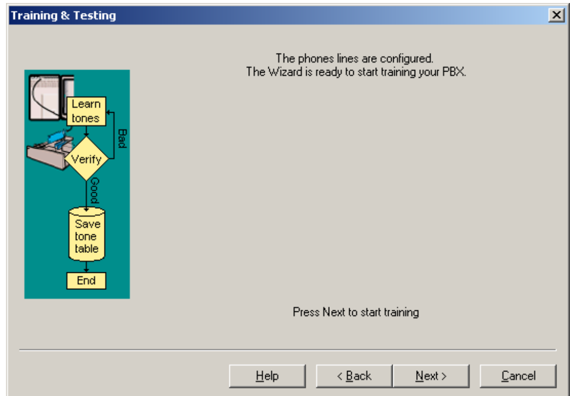
When you select Next, the AccuCall Wizard checks to make sure it can detect a phone line at the second extension. If the AccuCall Wizard does not detect a line, an error message appears that says: No loop current detected on first channel. Make sure the line is properly connected. Ensure that the phone line is connected at both ends. You can also plug an ordinary phone into the extension to check if the socket is working.

The AccuCall Wizard calls between the two extensions to verify the extension numbers. If either extension is incorrect, the AccuCall Wizard displays an error message indicating which of the two numbers was not correct. You can change to the Second Extension dialog box or select Back to re-enter the number for the First Extension dialog box.

You cannot proceed to the Training & Testing dialog box until the AccuCall Wizard verifies both extension numbers.

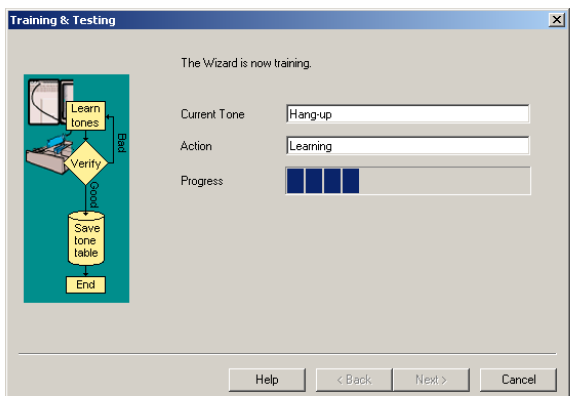
Training and Testing Window

This Training and Testing window informs you that the AccuCall Wizard is ready to train the tones you have specified.



Select Next to train the tones.

The window changes to inform you about the AccuCall Wizard training status.



The Learning the Hangup Tone dialog box displays the current status of the AccuCall Wizard as it trains. It provides the following status items:

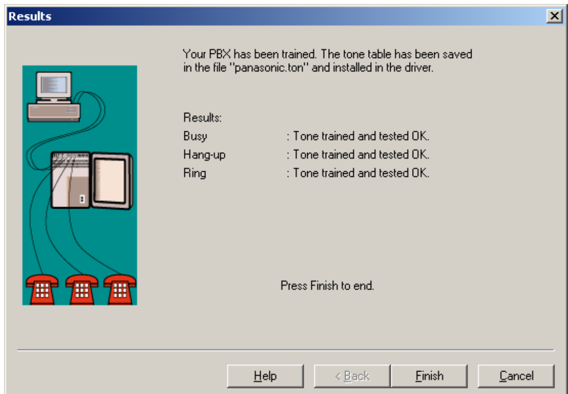
Current Tone	Displays the name of the tone the AccuCall Wizard is training.
Action	Displays the action the AccuCall Wizard is performing as it trains the tone.
Progress	Displays progress as a progress bar.
Next, Back, Cancel	The Next and Back buttons are disabled while the AccuCall Wizard trains the selected tones.

You can press Cancel at any time, but it might take up to 10 seconds for the AccuCall Wizard to stop. Please wait.

When the AccuCall Wizard finishes training, the Results dialog box appears.

Results Window

The Results window displays the results of the training session for each tone.



If the AccuCall Wizard failed to train any tones, it creates the ***accuwiz*** directory and saves the PCPM table along with a recording of all tones and log files.

You can fix a failed tone using the AccuCall utility, described in ***Using Dialogic® Brooktrout® AccuCall on page 206***, or you can contact Brooktrout Technical Support for help. Brooktrout uses the files in the ***accuwiz*** directory to diagnose the problem. See ***Getting Technical Support on page xxxvi***.

For each tone, the AccuCall Wizard displays one of the following messages shown in ***Table 9 on page 173***:

Table 9. Tone Messages

Message	Meaning
Failed to train tone	The AccuCall Wizard was unable to train this tone, but created a PCPM table containing the failed tone along with the other tones.
Tone trained but failed to save	The AccuCall Wizard was unable to save this tone to the PCPM table, but still made a PCPM table containing the other tones.
Tone trained but failed test	The tone failed the test, but the AccuCall Wizard created a PCPM table containing the failed tone along with the other tones.
Silence detected	The tone does not exist on your PBX, so the AccuCall Wizard detected silence. This is NOT an error message.
Tone trained and tested OK	The AccuCall Wizard trained the tone and added it to the PCPM table.

Select Finish to end the AccuCall Wizard session.

Troubleshooting

The AccuCall Wizard saves the recordings of any tone it failed to train in the ***accuwiz*** directory. You can send this file to Technical Support for help or use the AccuCall utility to fix the PCPM table See either:

- ***Using Dialogic® Brooktrout® AccuCall on page 206***
- ***Getting Technical Support on page xxxvi***

Custom Tones

This section gives examples of how to configure custom tones in the AccuCall Wizard Custom Tone Configuration dialog box. See ***Custom Tone Configuration Dialog Box on page 168***.

Preconfigured Tones

The AccuCall Wizard automatically trains the preconfigured Ring, Busy, and Hangup tones. You can view the configuration for these tones by selecting the Configure button next to any of the three tones in the Advanced Settings dialog box.

The following examples illustrate how to configure the three default tones. In these examples, [EXT0/1] indicates the number of either extension 0 or 1.

Ring

Call Progress Code

CS_RNGNOANS

Terminating

Unchecked.

Line Setup

Function	***Channel***	***Dial String***
Dial	0	[EXT1]

This configuration tells channel 0 to dial channel 1. Channel 0 then hears the Ring tone.

Busy

Call Progress Code

CS_BUSY1

Terminating

Checked.

Line Setup

Function	***Channel***	***Dial String***
Offhook	1	
Dial	0	[EXT1]

This configuration tells channel 1 to go offhook. Channel 0 then dials channel 1 and hears the busy tone.

Hangup

Call Progress Code	CS_ROBUSY		
Terminating	Checked.		
Line Setup	***Function***	***Channel***	***Dial String***
	Dial	1	[EXT0]
	Wait for call	0	
	Off hook	0	
	On hook	1	

This configuration tells channel 1 to dial channel 0. Channel 0 waits for the call, then goes offhook. Then channel 1 goes onhook and channel 0 hears the Hangup tone.

Some PBXs do not use a hangup tone. In this case, a message appears in the AccuCall Wizard final window saying that silence was detected.

Configuration Examples

The following examples show the configuration settings for common call progress tones. In each of these examples, [EXT0/1] indicates the number of either extension 0 or 1.

Dial Tone

For channel 0 to hear the dial tone used by your PBX, use the following settings:

Call Progress Code	CS_DIALTON		
Terminating	Unchecked.		
Line Setup	***Function***	***Channel***	***Dial String***
	Offhook	0	

This configuration tells channel 0 to go offhook and hear the dial tone.

Network Ring Tone

For channel 0 to hear the external ring tone used by your PBX, use the following settings:

Call Progress Code	CS_RNGNOANS		
Terminating	Unchecked		
Line Setup	***Function***	***Channel***	***Dial String***
	Dial	0	[and non-busy number]

This configuration tells channel 0 to dial any non-busy number and hear the network ring tone.

Network Busy Tone

For channel 0 to hear the external busy tone used by your PBX, use the following setting:

Call Progress Code	CS_BUSY1		
Terminating	Checked		
Line Setup	***Function***	***Channel***	***Dial String***
	Dial	0	[any busy number]

This configuration tells channel 0 to dial a busy number and hear the external busy tone.

Reorder Tone

The reorder tone, also called a fast busy, is the tone your PBX produces when all of its lines are busy. For channel 0 to hear the reorder tone used by your PBX, use the following settings:

Call Progress Code	CS_ROBUSY		
Terminating	Checked		
Line Setup	***Function***	***Channel***	***Dial String***
	Dial	0	[any invalid number]

This configuration tells channel 0 to dial an invalid number and hear the reorder tone.

Understanding Dialogic® Brooktrout® AccuCall

This section describes how to use Dialogic® Brooktrout® AccuCall menus, tools, and commands:

- ***Working With PCPM Tables on page 177***
- ***Tones on page 180***
- ***Understanding Filters on page 184***
- ***Start Parameters on page 190***
- ***Deglitching Parameters on page 192***
- ***Answer Parameters on page 195***
- ***Using the Toolbar on page 198***
- ***Accessing the Tools Menu on page 199***
- ***Accessing the Tools Menu on page 199***
- ***Loading to the Module on page 204***
- ***Setting Up a Call on page 204***
- ***Getting Help on page 205***

Working With PCPM Tables

To begin, open an existing PCPM table using the Open command in the File menu, or create a new PCPM table using the New command in the File menu.

Five folder tabs appear when you start AccuCall:

- ***Tones on page 180***
- ***Understanding Filters on page 184***
- ***Start Parameters on page 190***
- ***Deglitching Parameters on page 192***
- ***Answer Parameters on page 195***

Use the View menu to select one of these. The View menu also controls whether the toolbar and the status bar are displayed or hidden. For more information on these features, see ***Using the Toolbar on page 198*** and ***Accessing the Tools Menu on page 199***.

Tones Tab

The Tones tab displays all named tones in the currently open PCPM table and allows you to modify these tones or create a new table using the following buttons:

Edit

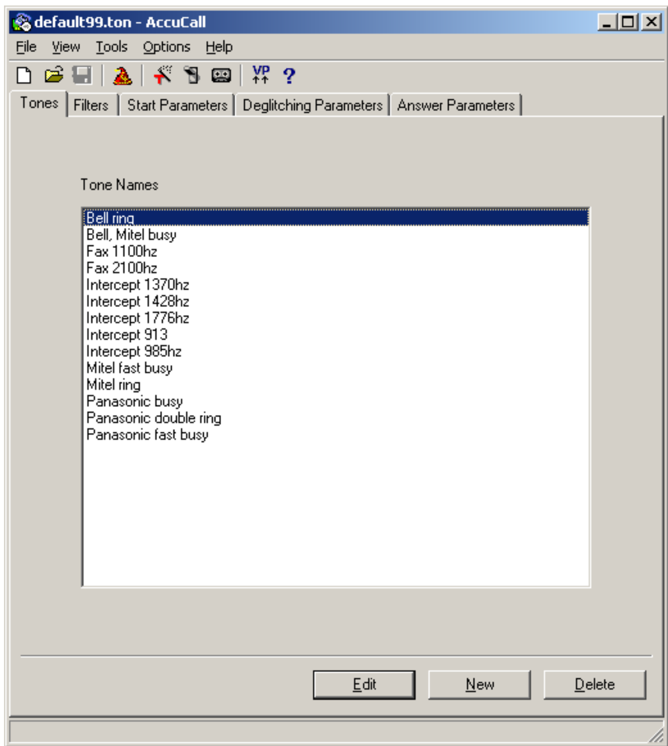
Displays the Edit Tone dialog box for a highlighted tone. You can double-click on a tone to accomplish the same thing.

New

Displays the Edit Tone dialog box to create a tone by entering its characteristics manually. All of the values in the display are blank or set to zero. You cannot test or install a PCPM table unless it has at least one tone saved in it.

Delete

Removes a tone permanently from the PCPM table after a confirmation box appears.



The default PCPM table contains definitions for several different call progress tones. See ***Call Progress Tones on page 159*** for an overview of basic PCPM concepts.

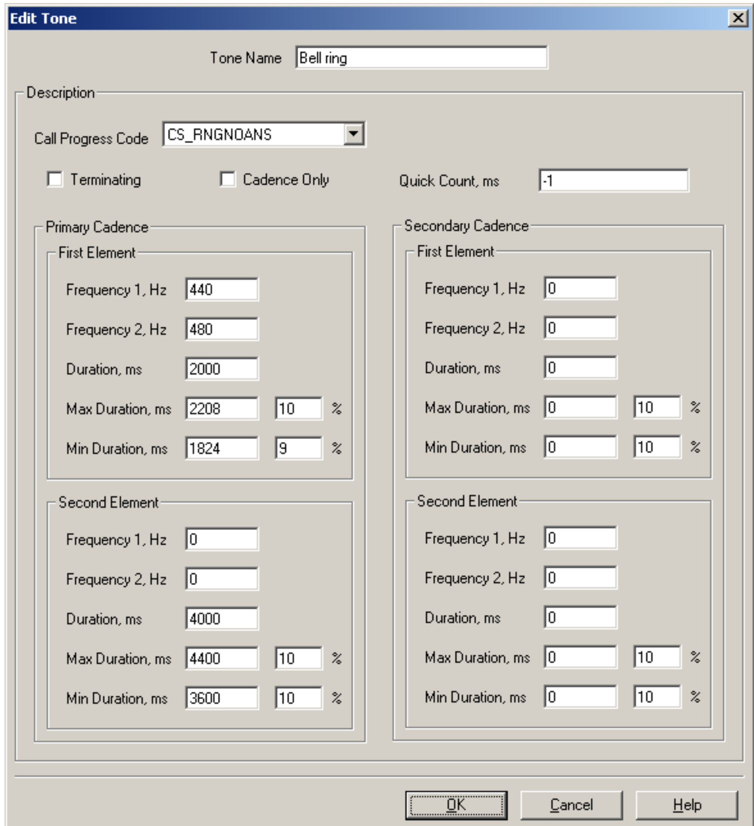
The definitions in the default PCPM table are for specific tones from specific PBXs. Since no one PCPM table can include all possible tones, AccuCall helps you customize a PCPM table to meet your requirements.

The default PCPM table also includes settings for the Start Parameters, Deglitching Parameters, Answer Parameters, and filter characteristics (Minimum Energy, Inband Ratio, Dualtone Ratio, Outband Ratio, WideBand Energy). You might need to alter these parameters to improve detection, but begin with the default settings.

Tones

Edit Tone Dialog Box

The Edit Tone dialog box appears when you click Edit or New button on the Tones tab.



Tone Characteristics

Make modifications by selecting, checking or typing over existing field values. Click OK to save your changes.

Tone Name

Displays the name of your tone. This can include up to 31 alphanumeric characters.

Call Progress Code

Value to report when this tone is detected. Use for call progress analysis. See ***Custom Tones on page 174***.

Terminating	Selects whether the tone is terminating. A terminating tone causes the board to terminate a multi-stage function such as play or record within a call. A multi-stage function executes in stages, as it receives event notifications. Terminating tones can use only real-time filters. See ***Understanding Filters on page 184*** for more information. An example of a terminating tone is a hang-up tone. If an application is playing a recording to a caller and the caller hangs up, a hang-up tone is played on the line. The application can detect the tone and can terminate the playing (stopping the playback is an application action item) only if the hang-up tone is identified in the PCPM table as a terminating tone.
Cadence Only	Selects whether AccuCall recognizes a tone by its cadence alone, rather than by its frequency and cadence together. If the box is checked, your board recognizes the tone only by its cadence. See ***Elements on page 182*** for more information.
Quick Count	Selects whether AccuCall recognizes a tone by its frequency alone, rather than by its frequency and cadence together. If the tone frequencies are on the line for the time specified by Quick Count, AccuCall detects the tone. Quick Count is useful for tones that do not have a regular cadence, such as pager tones. It can also speed up detection in certain situations. Once you set Quick Count to a value greater than 0, AccuCall sets all the Durations in the tone equal to Quick Count and grays them out. The Durations are no longer relevant for detecting the tone. Enter -1 to disable Quick Count. The default is disabled. See ***Improving Tone Detection on page 219*** for more information on Quick Count.

Elements

A segment is a continuous period of sound with one or two main frequencies. A segment can also be a period of silence, in which case the frequencies are 0.

An element is a segment with a fixed duration that belongs to a tone in the PCPM table. AccuCall constantly receives 16 ms samples from the board. It tries to patch them together into samples and match them to elements in the PCPM table. A tone in the tone table can contain up to four elements. These four elements define the tone's frequencies and cadences.

A cadence is a repeating sound consisting of two elements joined together. Many call progress tones have only one on/off cadence. For example, a Bell ring tone generates a 440 / 480 Hz sound for 2 seconds (the first element) and falls silent for 4 seconds (the second element). The pattern then repeats.

AccuCall can identify tones with any number of cadences or even no regular cadences at all. However, the PCPM table is structured around tones with one or two cadences. AccuCall must use frequency information alone to identify more complex tones.

The following fields describe the frequencies and duration of each element in the tone. In these fields, AccuCall rounds any entry to the nearest multiple of 16 ms.

Frequency 1 & 2
: Sets the element's frequencies. If the element is single-frequency, set Frequency 2 to 0. If the element is silent, set both fields to 0. If the element is dual-frequency (like the Bell ring tone described earlier), both fields need entries.

Duration
: Sets the duration for the element.

Max Duration
: Sets the maximum duration for the element.

Min Duration
: Sets the minimum duration for the element.

For cadences that contain an element of silence, add 32 ms to the desired Duration of the element containing frequencies and subtract 32 ms from the desired Duration of the element containing silence. As tones transition from a silent state to an on state, in most cases, approximately 32 ms of the tones silence period is detected as on.

If the tone was trained with AccuCall Wizard, this adjustment is automatically performed.

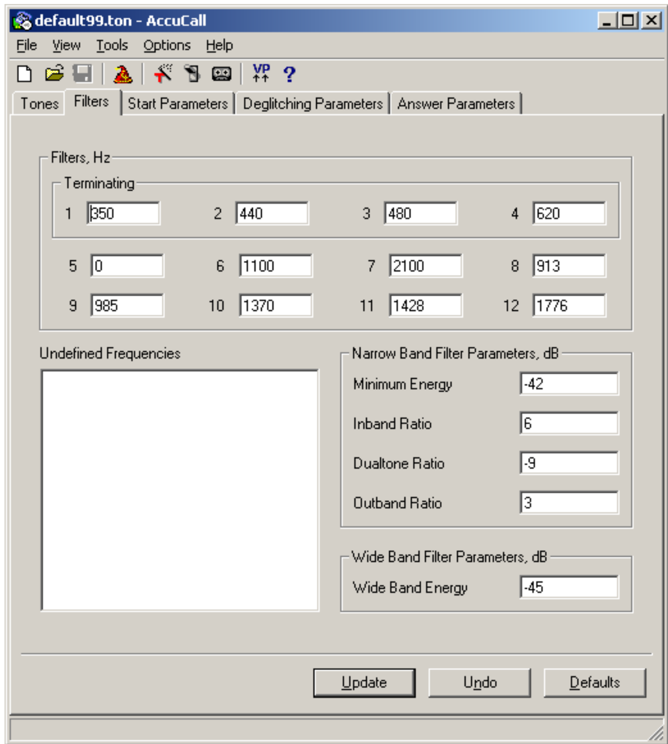
If an unidentified segment does not fall within the given Duration tolerances, AccuCall cannot recognize the element. Ensure that Max Duration and Min Duration are not too close to Duration. However, if these tolerances are too large, AccuCall can misidentify tones. For an example of this problem, see ***Demonstrating Cadence Overlap on page 220***.

Understanding Filters

Twelve filters on your board determine which frequencies are processed.

Filters Tab

In the Filters tab, you can change these acceptable frequencies and adjust filter characteristics to improve tone detection.



Each filter is a ±20 Hz window centered on the desired frequency. The nominal values of these filters determine the frequency components of the sample. The frequency components are all the sound energy that passes through the ±20 Hz window. Each tone in your PCPM table needs an open filter for each of its frequencies.

When the filter center frequency is more than a few Hz away from a frequency in a tone, sampling errors increase dramatically. For this reason, if a tone in the PCPM table does not have a filter setting that matches exactly, AccuCall puts the unmatched frequency in the Undefined Frequencies box.

Four filters are dedicated as terminating tone filters (also called real-time filters, because these filters are always active, whereas other filters are active only during PCPM). The PCPM algorithm uses these filters to identify tones when a multi-stage function is running. This is called background PCPM. If background PCPM is running during a multi-stage function, your application can recognize certain tones and terminate the function.

An example of background PCPM occurs when a caller hangs up while listening to a recorded message. If the application is running background PCPM, it can recognize the hang-up tone ***while it is playing the recorded message*** and terminate the play function. In non-background PCPM (also called outbound PCPM), the application is merely listening for tones or voice to detect a call answered condition. It is not running a multi-stage function at the same time.

You can use all twelve filters for non-terminating tones.

The Filters tab has the following fields and functions:

Filters 1-4

Sets frequencies for terminating (real-time) filters. The frequency can be 0, or can range from 300 – 3700 Hz. 0 means the filter is not open to any frequency.

Filters 5-12

Sets frequencies for non-terminating filters. The frequency can be 0, or can range from 300 – 3700 Hz. 0 means the filter is not open to any frequency.

Undefined Frequencies

Displays frequencies in your PCPM table that do not have filters assigned to them. If you do not assign filters to those frequencies, you cannot install your PCPM table to the driver, as described in ***Loading to the Module on page 204***.

Update

Updates your PCPM table with the current filter settings.

Undo

Restores filter settings 1 – 12 and all filter characteristics to the values recorded in the previous PCPM table update.

Defaults

Restores the five filter characteristics to their default values.

Setting Filter Characteristics

AccuCall tries to join raw 16 ms samples that have the same frequency components into longer segments. Before it can do this, it must determine the basic structure of each sample. Is it a dual-frequency sample, a single-frequency sample, or something else?

AccuCall ranks the amplitudes of the sample's frequency components from strongest to weakest. It then uses this information to determine the sample structure. You can adjust this process by modifying filter characteristics.

All of the filter characteristics are measured in units called dB. This is a logarithmic, absolute scale for measuring signal energy. A change of 3 dB corresponds ***exactly*** to a factor of 2. Do not confuse dB with conventional engineering units such as decibels or dBm.

Narrow Band Filter Characteristics

These characteristics involve the strengths of the sample's frequency components, as determined by the nominal values of the twelve filters.

Minimum Energy

Sets the smallest energy level that a narrow band filter detects as a signal. The default is –42 dB.

Inband Ratio

Sets the minimum acceptable inband ratio.

The inband ratio test helps determine whether a sample is single-frequency, dual-frequency, or neither. The test compares the strength of the weakest frequency that ***might be*** part of the sample to the frequency just below that, presumably ***not*** part of the sample. It is really two tests in one:

- The test compares the strongest frequency in the sample with the next strongest. In a good ***single***-frequency tone, the difference should be large.
- The test compares the second-strongest frequency in the sample with the third strongest frequency. In a good ***dual***-frequency tone, the difference should be large.

The test becomes more tolerant as you lower the dB value (meaning that a lower-quality signal is more likely to pass the test). The default is 6 dB.

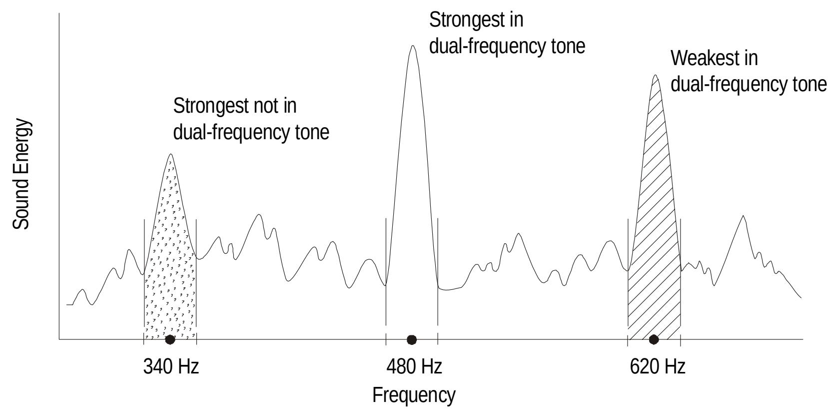


Figure 2. Inband Ratio

Figure 2 represents half of the inband ratio test graphically. In an ideal dual-frequency sample, the hatched area (energy in the 620 Hz filter) is much greater than the dotted area (energy in the 340 Hz filter). The other half of the test compares the 480 Hz peak to the 620 Hz peak.

Dualtone Ratio

Sets the maximum acceptable dualtone ratio.

The dualtone ratio test helps determine whether or not the sample is dual-frequency. The test compares the strongest frequency in the signal to the second-strongest frequency in the signal. In a good dual-frequency tone, the difference should be small.

The test becomes more tolerant as you lower the dB value. The default is –9 dB.

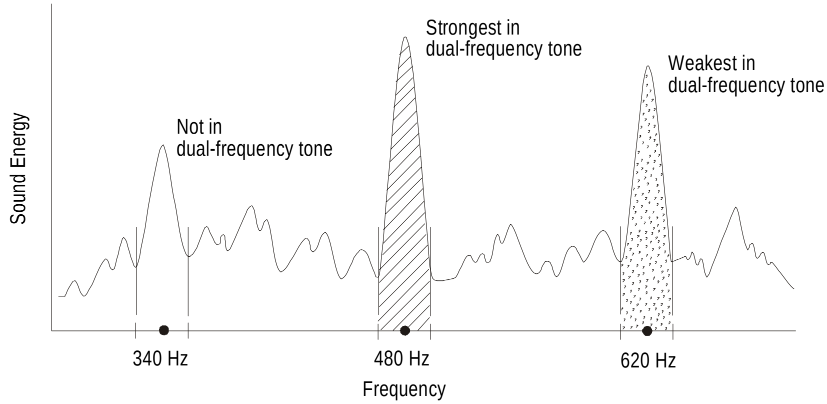


Figure 3. Dual-Frequency Tones

Figure 3 illustrates the dualtone ratio test graphically. In an ideal dual-frequency tone, the hatched area and the dotted area are about the same size.

Outband Ratio

Sets the minimum acceptable outband ratio.

The outband ratio test helps determine two things: whether the line is noisy, and whether the signal has frequencies for which you have no open filters. The outband ratio is really two tests in one:

- It compares the strongest frequency to all other sound on the line. In a good single-frequency tone, the difference should be large.
- It compares the sum of the strongest frequency and the second-strongest frequency to all other sound on the line. In a good dual-frequency tone, the difference should be large.

The test becomes more tolerant as you lower the dB value. The default is 3 dB.

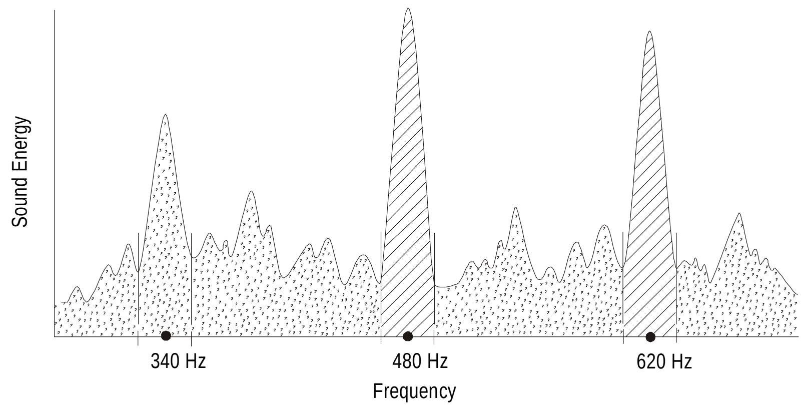


Figure 4. Outband Ratio

Figure 4 represents half of the outband ratio test graphically. In an ideal dual-frequency tone, the hatched area is much greater than the dotted area.

Wide Band Filter Characteristics

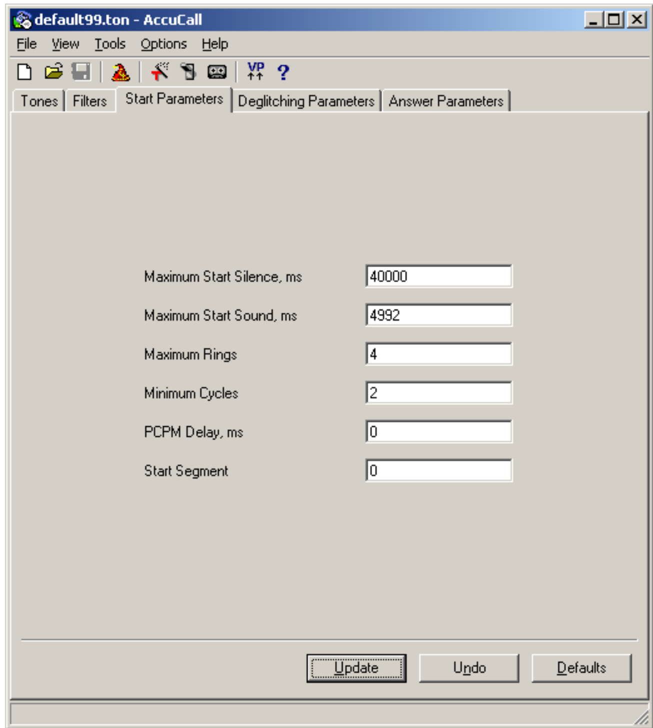
Wide Band Energy

Sets the minimum energy required for the wide band filter to detect a signal. The wide band filter passes frequencies from 300 – 3700 Hz. The default is –45 dB.

Start Parameters

Start Parameters Tab

Start Parameters control how PCPM begins its analysis. See *Call Progress Tones on page 159* for an overview of basic PCPM concepts.



Maximum Start Silence

Sets the maximum duration of a silence period at the start of outbound PCPM. If the silence period exceeds Maximum Start Silence, the signal is unidentified and PCPM terminates.

Set Maximum Start Silence to be greater than both Signal Dropout and the longest silence element of any tone in the PCPM table. The range is 16 ms – 8.7 minutes, in 16 ms increments. The default is 40 seconds.

Maximum Start Sound

Sets the maximum duration of a sound period at the start of outbound PCPM. If the sound period exceeds Maximum Start Sound, the signal is unidentified and PCPM terminates.

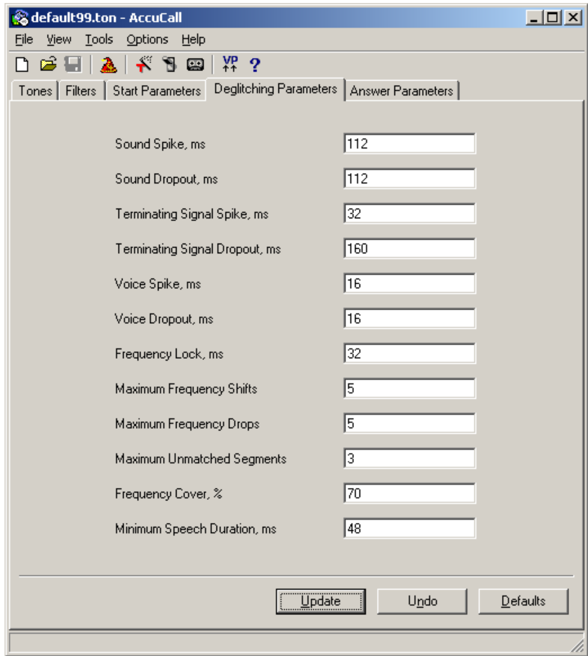
Set Maximum Start Sound to be greater than both Signal Spike and the longest sound element of any tone in the PCPM table. The range is 16 ms – 8.7 minutes, in 16 ms increments. The default is 2000 ms.

Maximum Rings	Sets the maximum number of ringback cycles. After detecting this many ringback cycles, PCPM considers the call unanswered and returns the Call Progress Code for the tone. PCPM needs to wait several cycles before terminating a ringback tone, to give the called party a chance to answer. The range is 0 – 255 cycles. The default is 4 cycles. AccuCall Wizard has a default of 6 cycles.
Minimum Cycles	Sets the minimum number of non-ringback cycles. PCPM must detect this many non-ringback cycles before it can return the Call Progress Code for the tone. Unlike ringback tones, PCPM needs to respond to non-ringback tones as quickly as possible. The range is 1 – 255 cycles. The default is 2 cycles.
PCPM Delay	Sets the time the outbound PCPM waits before it begins analyzing tones. This delay allows the signal on the line to stabilize after the connection is made. This can be necessary because some older switches generate spurious signals or clicks before generating the ringback tone. The range is 0 to 17.5 minutes, in 16 ms increments. The default is 0 ms (no delay).
Start Segment	Sets the number of deglitched segments that PCPM waits to receive before it starts analyzing tones. A deglitched segment is a period of silence/sound on the line that is longer than Sound Dropout/Sound Spike. Start Segment prevents detection problems caused by some older switches that generate spurious signals during the ringback tone. However, Start Segment can interfere with the detection of answer condition or frequency-only tones, so do not set it greater than zero unless absolutely necessary. If Start Segment is greater than zero, PCPM counts the first sound segment as the first segment. This rule eliminates the uncertainty of whether there is a segment of silence preceding the first tone. The range is 0 – 255 segments. The default is 0 (start PCPM immediately).
Update	Updates your PCPM table with the current filter settings.
Undo	Restores filter settings 1–12 and all filter characteristics to the values recorded in the previous PCPM table update.
Defaults	Restores the five filter characteristics to their default values.

Deglitching Parameters

Deglitching Parameters Tab

Deglitching Parameters control how PCPM deglitches tone elements. Deglitching is the process of removing spurious signals from a tone.



Signal Spike (SoundSpike)

Sets the minimum length of a non-voice signal. The non-voice signal must be longer than this, or PCPM considers it a noise spike and replaces it with silence.

Set Signal Spike to be less than Frequency Lock, Maximum Start Sound, and the duration of all sound elements in the PCPM table (except for elements of frequency-only tones). The range is 16 – 4080 ms in 16 ms increments. The default is 112 ms.

Signal Dropout (SoundDropout)

Sets the minimum duration of silence. The period of silence must be longer than this, or PCPM considers it a temporary dropout of signal and replaces it with the last sound detected.

Set Signal Dropout to be less than both Maximum Start Silence and the duration of all silence elements in the PCPM table. The range is 16 – 4080 ms in 16 ms increments. The default is 112 ms.

Terminating Signal Spike (TermSpike)

Sets the minimum length of a non-voice signal during background PCPM. The signal must be longer than this, or PCPM considers it a noise spike and replaces it with silence.

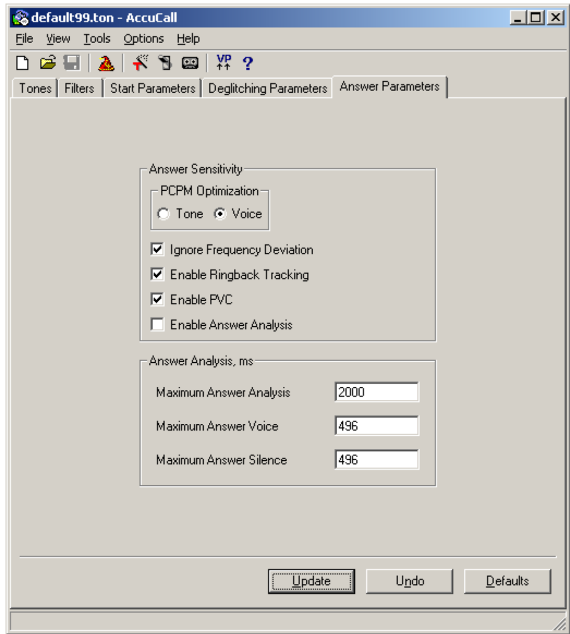
	Set Terminating Signal Spike to be less than the duration of all terminating sound elements in the PCPM table. The range is 16 – 4080 ms in 16 ms increments. The default is 32 ms.
Terminating Signal Dropout (TermDropout)	Sets the minimum duration of silence during background PCPM. The silence period must be longer than this, or PCPM considers it a temporary dropout of signal and replaces it with the last sound detected. Set Terminating Signal Dropout to be less than the duration of all terminating silence elements in the PCPM table. The range is 16 – 4080 ms in 16 ms increments. The default is 160 ms.
Voice Spike (PvcSpike)	Sets the minimum length of voice. The signal must be longer than this, or PCPM considers it a noise spike and adds its duration back to the absence of voice that was detected before. Set Voice Spike to be less than Minimum Speech Duration. The range is 16 – 4080 ms in 16 ms increments. The default is 16 ms.
Voice Dropout (PvcDropout)	Sets the minimum duration of absence of voice. If the absence of voice is shorter than this, PCPM considers it a dropout and adds it back to the voice signal that was detected before. The range is 16 – 4080 ms in 16 ms increments. The default is 16 ms.
Frequency Lock	Sets the time that a frequency must be detected to be considered a segment's main frequency. PCPM then uses that frequency to analyze a segment's characteristics. Background PCPM uses Terminating Signal Spike instead of Frequency Lock. Set Frequency Lock to be less than Signal Spike. It should also be less than the duration of all elements in the PCPM table that are not in cadence-only or frequency-only tones. If not, PCPM can report false call answered conditions. The range is 16 – 4080 ms in 16 ms increments. The default is 32 ms.
Maximum Frequency Shifts	Sets the maximum number of times a frequency can shift before PCPM considers the call answered. A frequency shift occurs when a signal suddenly changes frequencies. Every time this happens, PCPM increments a frequency shift counter. When PCPM deglitches a new segment, it sets the frequency shift counter back to 0. If the counter ever exceeds Maximum Frequency Shifts, PCPM considers voice present on the line, which means the call is answered. Maximum Frequency Shifts functions only if PCPM is optimized to Voice. The range is 0 – 255 frequency shifts. The default is 5.

Maximum Frequency Drops	Sets the maximum number of times a frequency can drop before PCPM considers the call answered. A frequency drop occurs when a signal suddenly falls silent. Every time this happens, PCPM increments a frequency drop counter. When PCPM deglitches a new segment, it sets the frequency drop counter back to 0. If the counter ever exceeds Maximum Frequency Drops, PCPM considers voice present on the line, which means the call is answered. Maximum Frequency Drops functions only if PCPM is optimized to Voice. The range is 0 – 255 frequency drops. The default is 5.
Maximum Unmatched Segments	Sets the maximum number of segments that do not match any elements in the PCPM table before PCPM considers the call answered. If PCPM detects more unmatched segments than this, it considers voice present on the line, which means the call is answered. Maximum Unmatched Segments functions only if PCPM is optimized to Voice. The range is 0 – 255 unmatched segments. The default is 3.
Frequency Cover	Sets the minimum percentage of a non-silence segment that must contain the same frequency. If the segment's main frequency covers less than this percentage of the total segment, PCPM considers the call answered. Frequency Cover functions only if PCPM is optimized to Voice. The range is 0 – 100%. The default is 70%.
Minimum Speech Duration	Sets the minimum duration of voice before PCPM considers the call answered. Set Minimum Speech Duration to be greater than Voice Spike; otherwise, PCPM might consider noise spikes to be a call-answered condition. The range is 0 to 17.5 minutes, in increments of 16 ms. The default is 48 ms.
Update	Updates your PCPM table with the current filter settings.
Undo	Restores filter settings 1 – 12 and all filter characteristics to the values recorded in the previous PCPM table update.
Defaults	Restores the five filter characteristics to their default values.

Answer Parameters

Answer Parameters Tab

The Answer Parameters control how PCPM deals with call answered conditions.



Answer Sensitivity pane

The Answer Sensitivity pane parameters affect how often PCPM reports that calls have been answered.

PCPM Optimization

Selects whether PCPM is optimized for voice or tone detection. Select Voice to have PCPM consider most deviations from a clear signal to be voice. This makes detection more sensitive to voice. However, if the line is noisy or if cadence-only tones exist in the PCPM table, select Tone. When set to Tone, PCPM is less sensitive to voice, and does not use the parameters Maximum Frequency Shifts, Maximum Frequency Drops, Maximum Unmatched Segments, and Frequency Cover. The default setting is Voice.

Ignore Frequency Deviation

Selects whether PCPM ignores small frequency deviations that do not satisfy the Inband Ratio, Outband Ratio, or Dualtone Ratio filter parameters. Enabling Ignore Frequency Deviation is similar to dynamically adjusting the filter parameters. If you disable Ignore Frequency Deviation, PCPM considers these deviations as frequency shifts, which makes PCPM much more sensitive to line noise and can lead to false call answered reports. By default this parameter is enabled.

Enable Ringback Tracking	Selects whether PCPM should detect the call answered condition by monitoring changes in ringback tones. Enable Ringback Tracking makes PCPM report a call answered condition if there is any change from a ringback tone: change in cadence, break in cadence, or change in frequency. This option speeds up answer detection. If Enable Ringback Tracking is not selected, PCPM waits for events such as voice detection or unmatched segments before reporting the call answered. Enable Ringback Tracking functions only during outbound PCPM. The default setting is enabled.
Enable PVC	Selects whether Brooktrout's proprietary Positive Voice Control (PVC) algorithm should be used to detect voice. PVC is a separate voice detection algorithm that runs in parallel with PCPM. Selecting Enable PVC allows faster and more reliable detection of call answered conditions. The default setting is enabled.
Enable Answer Analysis	Selects whether to analyze answer characteristics, which allows an application to identify various greetings, such as those spoken by a live person, a receptionist, or an answering machine. The default setting is disabled.

Answer Analysis pane

The Answer Analysis pane parameters provide additional information that your application can use to identify different types of greetings. For example, a person at home might answer with a simple "Hello", a receptionist might say: "Good morning, XYZ Corp., how may I help you". An answering machine can have an even longer greeting. In this case, Maximum Answer Voice helps your application distinguish between the three.

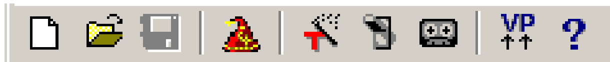
Maximum Answer Analysis	Sets the maximum duration to run answer analysis. If answer analysis runs for this amount of time without detecting the amount of voice or silence specified by Maximum Answer Voice or Maximum Answer Silence, it terminates. The range is 0 to 17.5 minutes, in 16 ms increments. The default is 2 seconds.
Maximum Answer Voice	Sets the maximum continuous amount of voice that answer analysis accepts before terminating. Set Maximum Answer Voice to be less than Maximum Answer Analysis. The range is 0 to 17.5 minutes, in 16 ms increments. The default is 496 ms.

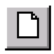
Maximum Answer Silence	Sets the maximum continuous amount of silence that answer analysis accepts before terminating. Set Maximum Answer Silence to be less than Maximum Answer Analysis. The range is 0 to 17.5 minutes, in 16 ms increments. The default is 496 ms.
Update	Updates your PCPM table with the current filter settings.
Undo	Restores filter settings 1 – 12 and all filter characteristics to the values recorded in the previous PCPM table update.
Defaults	Restores the five filter characteristics to their default values.

Using the Toolbar

AccuCall Toolbar

Select the Toolbar command in the View menu to display the toolbar at the top of the window, just below the menu bar. The toolbar provides quick mouse access to all of the major functions of AccuCall.





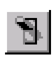
Creates a new PCPM table.

Opens a saved PCPM table.

Saves a PCPM table. This function is disabled if the PCPM table is recently saved or not modified since opening the PCPM table.

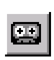
Activates AccuCall Wizard. For instructions on how to use AccuCall Wizard to train PCPM tables, see ***Dialogic® Brooktrout® AccuCall Wizard on page 159***.

Learns a tone.



Tests a tone. This function is disabled if either:

- There is no open PCPM table.
- Currently open PCPM table is not properly loaded to the driver.
- There is an error in the currently open PCPM table.



Records a tone.

Loads the PCPM table to the driver. This function is disabled if there is:

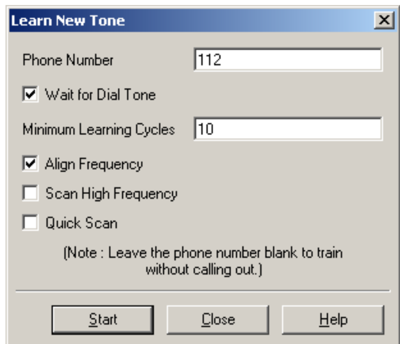
- No open PCPM table.
- An error preventing the loading (such as mismatched filter settings).

Accessing the Tools Menu

The Tools menu provides access to three basic functions in AccuCall: Learn New Tone, Test PCPM Table, and Record Tone.

Learn New Tone Dialog Box

Use the Learn New Tone dialog box to add tones to your PCPM table by listening to tones over a phone line.



Phone Number

Sets the phone number that carries the tone you want to add. If you are not calling out, leave the field blank.

Wait for Dial Tone

Selects whether to listen for a dial tone before making the call. This ensures that the board is connected to a working phone line.

Minimum Learning Cycles

Sets the minimum number of full cycles that AccuCall uses to learn the new tone.

Align Frequency

Selects whether to align the frequencies in the new tone to previously detected frequencies that are nearby (within 20 Hz). This option prevents frequencies that are shared in multiple tones from being characterized incorrectly by bad samples. For example, if a previously recorded tone had a component of 340 Hz, AccuCall would automatically adjust a new tone of 343 Hz to 340 Hz. This option is selected by default.

Scan High Frequency

Selects whether to scan frequencies from 800 – 1700 Hz. Select this option to scan for higher frequency signals such as pager tones. If not selected AccuCall scans frequencies from 300 – 1700 Hz

Quick Scan

Selects whether to scan for a brief amount of time instead of many cycles. Select this option to learn tones with no cadence, such as fax tones, or continuous tones (such as dial tones). Specify the scan time by changing the Quick Count field in the Edit Tone dialog box, as described ***Tones on page 180***.

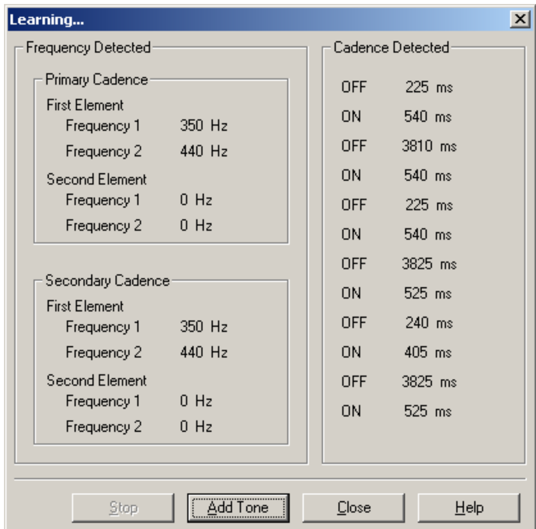
Start — Starts learning the tone.

Close — Closes the Learn New Tone dialog box without learning the tone.

Once you click Start, AccuCall calls the specified phone number and tries to add the tone it hears on the other end to the currently open PCPM table.

Learning... Window

The Learning... window contains many tone elements within the Frequency Detected, Secondary Cadence, and Cadence Detected panes. Use the buttons described below to perform actions with this window,



The following buttons appear in the Learning window:

Stop — Stops learning the tone. This button is available only while the tone is being learned.

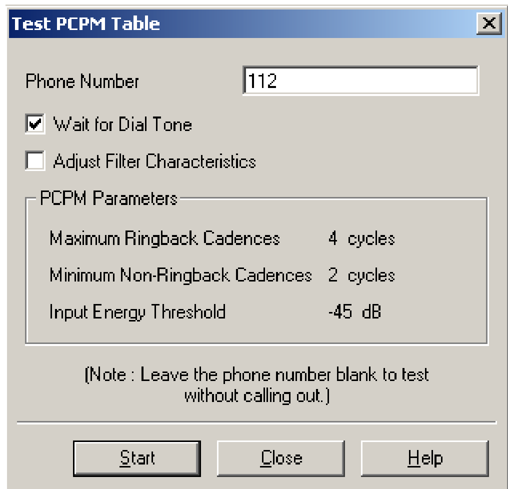
Add Tone — Adds the tone to your PCPM table. This button is available only after the learning process is complete.

Close — Closes the dialog box. This button is available only after the learning process is complete.

Test PCPM Table Dialog Box

This procedure tests whether the driver can identify a tone from your PCPM table. Once you create a new tone, or modify an existing tone, you need to test it.

There are three options that you can change in the Test PCPM Table dialog box. You can change the settings in the PCPM Parameters box only by selecting either the Start Parameters tab or the Filters tab. For more information on these parameters, see *Understanding Filters on page 184* and *Start Parameters on page 190*.



Phone Number

Specifies the number of the line that is to carry the tone you want to test against. Leave this field blank if you are not calling out to receive the tone source.

Wait for Dial Tone

Selects whether to listen for a dial tone before making the call. This option ensures that the board is connected to a working phone line. Check the box to select this option.

Adjust Filter Characteristics

Adjusts the filter characteristics (*Understanding Filters on page 184*) automatically if AccuCall is having trouble matching the tones. Check the box to select this option.

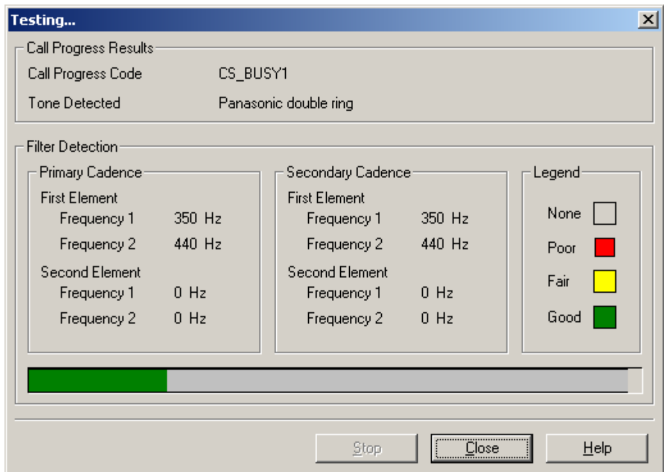
This option can improve tone detection a great deal. If you are having difficulty detecting tones, determine which tone is giving you the most trouble and test that tone with the Adjust Filter Characteristics box checked. If the filter settings change and you successfully detect the problem tone, uncheck the box, clear the rest of the PCPM table, and learn the rest of the tones with those filter settings.

Start	Starts testing the tone.
Close	Closes the Test PCPM Table dialog box without testing the tone.

When you click Start, the Testing dialog box appears.

Testing Dialog Box

The Testing Dialog box contains tone information within the following panes:



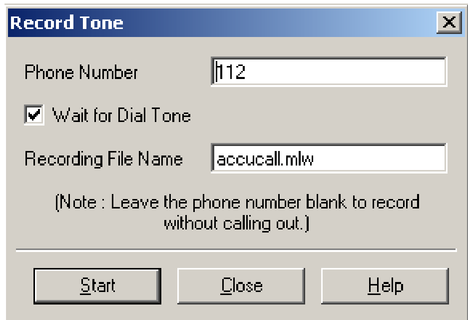
Call Progress Results Pane	This pane displays the Call Progress Code and the name of the tone detected. This information does not appear until the testing concludes.
Filter Detection Pane	This pane displays the elements of the detected tone and a bar that indicates the progress of the test procedure. AccuCall does not display the frequencies until testing concludes. A legend box indicates what the colors in the bar mean. Results of poor or fair mean that AccuCall is detecting many sampling errors.

If the bar is red and yellow, and testing continues for thirty seconds or more, click Stop. Check the Adjust Filter Characteristics and try again.

Record Tone Dialog Box

Use the Record Tone dialog box to record a suspect tone and examine it later, or send it to Technical Support for further analysis. The Record Tone dialog box generates a file in the µlaw PCM format (.mlw).

See ***Getting Technical Support on page xxxvi*** for more information.



Phone Number — Specifies the phone number that carries the tone you want to record. If you are not calling out, leave the field blank.

Wait for Dial Tone — Selects whether or not to listen for a dial tone before making the call. This option ensures that the board is connected to a working phone line. Check the box to select this option.

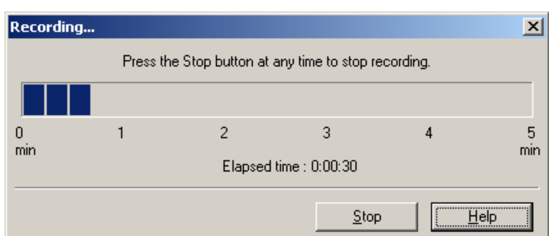
Recording File Name — Specifies the name of the file that AccuCall creates. The default file name is ***accucall.mlw***.

Start — Starts recording the tone.

Close — Closes the Record Tone dialog box without recording the tone.

Recording Dialog Box

Once AccuCall begins recording, the Recording dialog box appears. A blue progress bar indicates the duration of the recording. Click Stop to end recording.



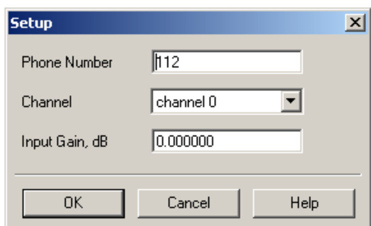
Loading to the Module

To load the current PCPM table to the voice processing driver, choose the Install Table to Module command from the menu. This works only if the PCPM table has no errors when you update it. For example, the PCPM table does not load if you failed to set an appropriate filter for a frequency in the PCPM table.

Setting Up a Call

You can select basic parameters such as the default telephone number or the active channel of your board by selecting the Setup command from the Options menu.

Setup Dialog Box



Phone Number

Sets the default telephone number. In any function that dials out, such as Learn New Tone, Test PCPM table, or Record Tone, this number appears first in the phone number field.

In addition to digits 0–9, you can enter the following characters into the Phone Number field:

Table 10. Phone Number Character Set

Code	Meaning
W	Waits for dialtone.
A B C D * #	These six characters are special DTMF digits.
,	The comma represents a pause for one second.
;	The semi-colon represents a pause for five seconds.

Table 10. Phone Number Character Set (Continued)

Code	Meaning
&	The ampersand represents a flash hook, used to start a call transfer. This feature is available only on analog boards.
T	Causes sequential digits to be dialed as DTMF.
P	Causes sequential digits to be dialed as Pulse.

Channel — Selects the channel to use during learning and testing.

Input Gain — Sets the amplitude of the incoming signal.

Do not adjust the Input Gain parameter without instruction from Technical Support. See ***Getting Technical Support on page xxxvi***.

Note: For users accustomed to previous versions of AccuCall, ensure that the flash hook duration is now set using the Dialogic® Brooktrout® Configuration Tool. (In the Bfv API, this value can still be set using AccuCall.)

Getting Help

Index — Displays help information, organized in an index.

Using Help — Displays information on how to use the help function.

About AccuCall... — Displays program information and version number.

Using Dialogic® Brooktrout® AccuCall

This section lists a few step-by-step procedures for basic functions in AccuCall. These procedures help you to get started. They are not meant to be complete.

- *Running Dialogic® Brooktrout® AccuCall on page 206*
- *Creating a Tone Manually on page 207*
- *Creating a Tone by Learning on page 210*
- *Comparing Tone Creation and Tone Learning on page 213*
- *Setting Filters for a New Tone on page 214*
- *Testing a Tone on page 216*
- *Recording a Tone on page 218*

Running Dialogic® Brooktrout® AccuCall

You can run the AccuCall utility from the file system, program menu, or the command line:

- File System:

  [INSTALLDIR]\utils\winnt\bin\AccuCall.exe
- Program Menu:

  Program → Brooktrout → AccuCall
- Command Line:

  c:\Brooktrout\boston\utils\winnt\bin\AccuCall.exe

Creating a Tone Manually

When you start AccuCall, you must either open an old PCPM table or create a new one. There are two ways of generating a tone for a PCPM table. You can have your board call a number and learn the tone there, or you can set the tone characteristics manually.

➢ ***This example describes creating a new PCPM table file and add a new tone:***

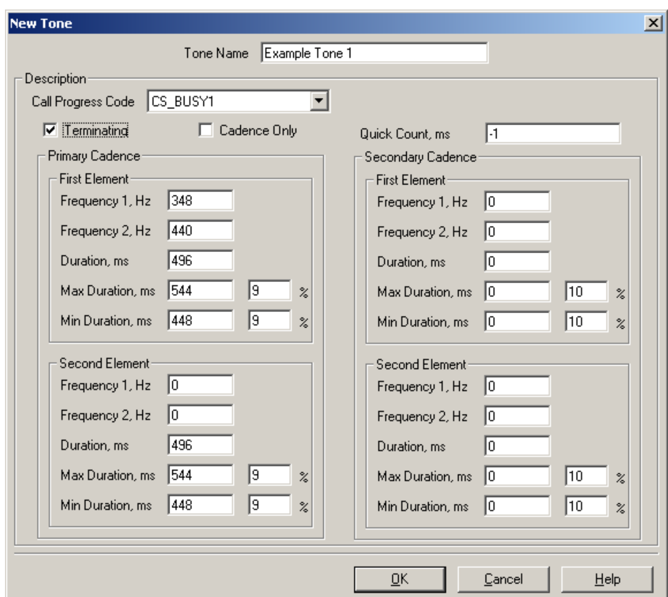
For more information on the Edit Tone dialog box, ***Tones on page 180***.

1. Create the PCPM table file by selecting the New command from the File menu.
2. Open the Edit Tone dialog box.

   Since your PCPM table is new, it contains no tones and the Tone Names window is blank. Click New to display the New Tone dialog box.
3. Enter a tone name.

New Tone Dialog Box

Your software and hardware refer to the tone by its Call Progress Code. The tone name is for your reference only, so use a descriptive, unique name that is easy to remember — like “Example Tone 1”.



4. Select the Call Progress Code. The following are Ringback tones:
   - CS_RING1
   - CS_RING2
   - CS_RNGNOANS

   If you select a Ringback tone, the driver uses the parameter Maximum Rings to decide how many tone cycles must be present before terminating PCPM. If you select a Non-Ringback tone, the driver uses Minimum Cycles. See ***Start Parameters on page 190***.

5. Decide whether the tone is terminating and/or cadence only.

   Check the Terminating box if the tone is meant to terminate a multi-stage function. See ***Tones on page 180***. Ringback tones cannot be terminating tones.

   Check the Cadence Only box if the signal is neither a single frequency tone nor a dual frequency tone, and should be recognized only by its cadence.

6. Enter the duration and frequencies of the first element in the primary cadence.

   Enter the duration and frequencies of the first element in the tone. If the tone is single-frequency, set Frequency 2 to 0. Make sure that Max Duration and Min Duration are not equal to Duration. You need to choose some level of tolerance, since sampling errors and noise distort the true duration. Note that all values you enter in Duration, Max Duration, and Min Duration are rounded to the nearest multiple of 16 ms.

   You need to know what the proper duration and frequencies are ahead of time. If you have access to the tone over a phone line, you can add it to your PCPM table automatically using AccuCall Wizard, described in ***Dialogic® Brooktrout® AccuCall Wizard on page 159***. You can also add it using the Learn New Tone function, described under ***Creating a Tone by Learning on page 210***.

   For cadences that contain an element of silence, add 32 ms to the desired Duration of the element containing frequencies, and subtract 32 ms from the desired Duration of the element containing silence. As tones transition from a silent state to an on state, in most cases, approximately 32 ms of the tones silence period is detected as on.

   If the tone was trained with AccuCall Wizard or by using the Learn New Tone function, this adjustment is automatically performed. Repeat this process on each of the elements in the primary cadence and secondary cadence, if one is provided.

7. Enter the duration and frequencies of the second element in the primary cadence.

   Enter the duration and frequencies of the second element in the tone. If the tone is single-frequency, set Frequency 2 to 0. If the second element is silent, set both Frequency 1 and Frequency 2 to 0.

   Again, if you do not know the characteristics of the tone, do a search for the tone, analyze a recorded tone, or add the tone to your table with the Learn New Tone function.

8. Repeat for the secondary cadence, if necessary.

   If the tone has only one cadence, ensure that Frequency 1, Frequency 2, Duration, Max Duration, and Min Duration in both elements are 0.

   Tones that have an irregular cadence or no cadence at all are called burst tones. See ***Optimizing Tone Detection Using Quick Count on page 228*** for a discussion of how to define burst tones using the Quick Count parameter.

9. Click OK to save your tone.

   If you have no filters assigned to the frequencies in the new tone, a dialog box appears. Click OK to automatically assign filters to the filters in the filter table. Click Cancel to automatically change the newly learned tone's frequencies to match the closest filters that are already open.

Creating a Tone by Learning

The second way to create a tone is to use the Learn New Tone function. This function creates the tone exactly as it is heard over the phone, and there is no need to look up what the tone is supposed to be.

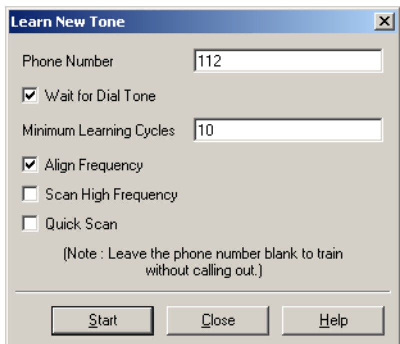
The simplest way to learn a tone is to call out to a line that is generating the desired signal. This example describes creating a new PCPM table, calling a line that is off hook, and providing the busy signal.

➢ ***Do the following to create a tone:***

1. Create the PCPM table file by selecting New in the File menu.

Learn New Tone Dialog Box

2. The Learn New Tone dialog box appears so you can set up the call.



Make sure that your board is properly connected to a line going to a Central Office (CO) or PBX. Use the Setup command in the Options menu to select the channel that you want to use. Connect that channel to a phone line. If you check the Wait For Dial Tone box, the board does not call out until it senses that the channel is physically connected to a phone line.

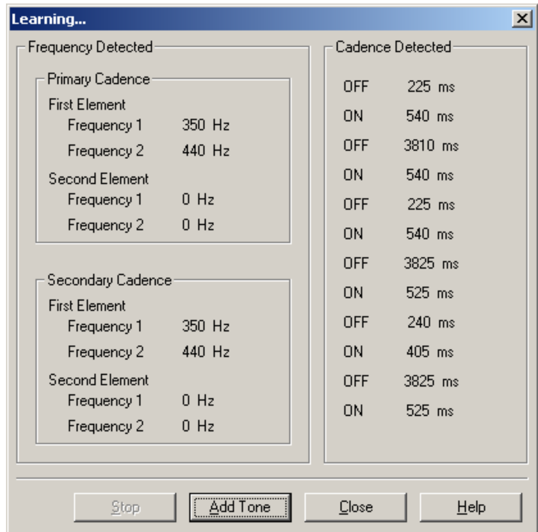
Enter the Phone Number that you want to call and make sure that the line you are calling is generating the desired signal. In this example, we are calling extension #112 on a PBX, and extension #112 is currently off hook.

Check the Align Frequency box so that AccuCall can match learned frequencies with previously detected frequencies. See ***Frequencies on page 224***.

3. Start learning the tone.

Learning Window

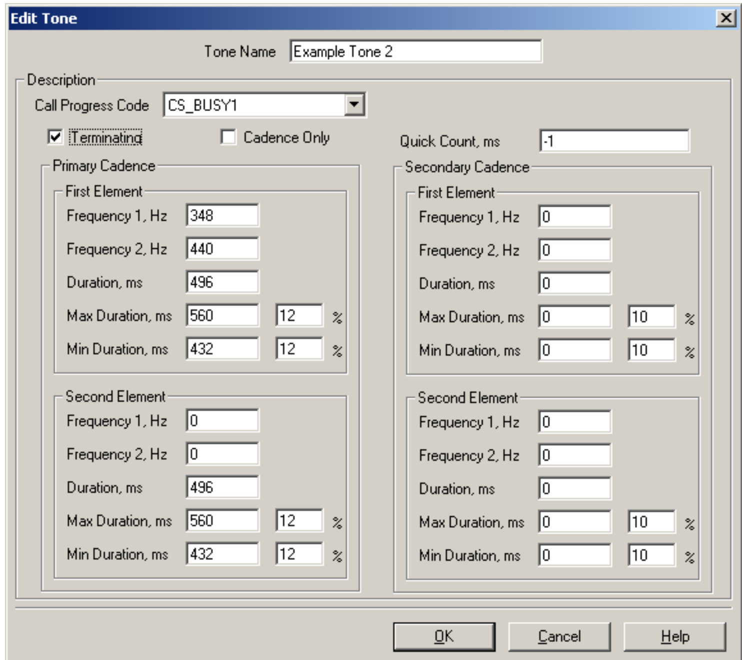
Click Start to begin the tone learning process.



AccuCall learns the tone by sampling over numerous cycles. You can halt the learning process by clicking Stop.

Edit Tone Dialog Box

Once AccuCall finishes, the frequencies of the tone are displayed at the top. Click the Add Tone button. The Edit Tone dialog box appears.



➢ ***Do the following to edit the tone characteristics.***

1. Give the tone a name.

   Your software and hardware refer to a tone by its Call Progress Code. The name is for your reference only, so use a descriptive, unique name that is easy to remember like: Example Tone 2.

2. Select the Call Progress Code. The following are Ringback tones:

   - CS_RING1
   - CS_RING2
   - CS_RNGNOANS

   If you select a Ringback tone, the driver uses the parameter for Maximum Rings to decide how many tone cycles must be present before terminating PCPM. If you select a Non-Ringback tone, the driver uses Minimum Cycles. See ***Start Parameters on page 190***.

3. Decide whether the tone is terminating and/or cadence only.

   Check the Terminating box if the tone is meant to signal the end of a multi-stage call. Examples of a terminating tone are a hang-up or a busy signal. Check the Cadence Only box if the signal is not a single or dual frequency tone, and recognized only by its cadence.

   At this point, you have finished creating the tone.

4. Click OK to save your tone.

   If you have no filters assigned to the frequencies in the new tone, a dialog box appears. Click OK to automatically assign filters to the filters in the filter table. Click Cancel to automatically change the newly learned tone's frequencies to match the closest filters that are already open.

Comparing Tone Creation and Tone Learning

When comparing ***Edit Tone Dialog Box on page 212*** to ***New Tone Dialog Box on page 207***, notice that while the frequencies are the same, the cadence is not.

Example Tone 1 was entered in by hand, with the durations of both elements set to 496 ms.

Example Tone 2 was learned: AccuCall set the durations and frequencies of the elements automatically. As it turns out, the true signal has an on time and an off time of 500 ms. This value falls within the tolerances of both Example Tone 1 and Example Tone 2. Both tones should test successfully.

Note: When AccuCall learns a tone that has an on/off cadence, the measured sound segment is always slightly longer than the real sound segment and the measured silence segment is always slightly shorter than the real silence segment. The samples that cover the sound/silence transition count as sound time rather than part sound and part silence. This occurs because the transition samples do have some sound energy, and AccuCall does not count them as silence. Usually the difference is about 32 ms.

Setting Filters for a New Tone

Your board provides twelve filters. Four of these are real-time filters, which are dedicated to terminating tones that function during background PCPM. You can use all twelve filters for non-terminating tones. Frequencies in a tone must first pass through one of these filters for PCPM to identify the signal.

AccuCall does not let you load the table to the driver unless every frequency in the PCPM table has an assigned filter frequency.

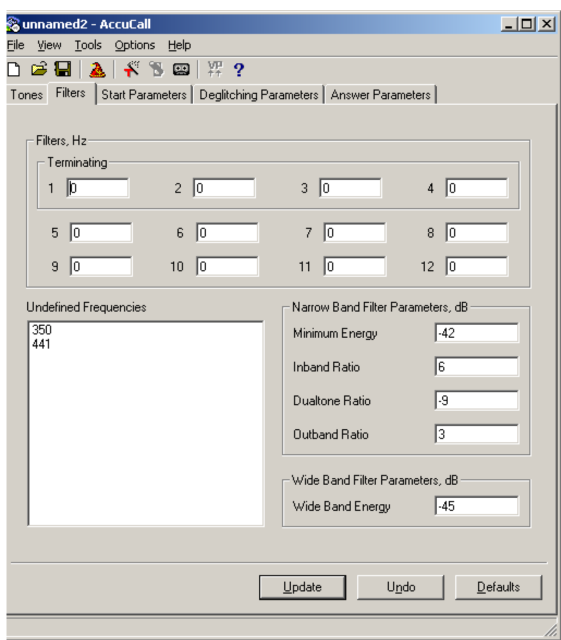
➢ ***Do one of the following to assign or edit frequencies:***

- If there are already some filters assigned to a nearby frequency, you can edit the tone so that its frequency matches the filter frequency. If you are only a few Hz off, this might be the only practical solution because you cannot assign a new filter within 40 Hz of an existing one. ***Defining Identical Frequencies on page 226*** and ***Dealing With 'Too-Close' Frequencies on page 234***.
- If there are no old filters within 40 Hz of the desired frequency, assign a filter to that frequency, as follows:

1. Go to the View menu and select the Filters tab.

Filters Tab

The Filters Tab appears.



This Filters tab shows the tone created in ***Creating a Tone by Learning on page 210***. There are two undefined frequencies.

2. Add the undefined frequencies.
3. Click Update.

   You have completed adjusting the filters.

Try changing the filter characteristics if you are having problems learning or testing tones. If the filter characteristics are not set correctly, bad samples can appear that interfere with testing and learning. To adjust these parameters automatically, use the option Adjust Filter Characteristics in the Test dialog box.

For example, when AccuCall tries to detect a tone on a very noisy line. If the outband ratio setting is too high, AccuCall does not recognize the tone, because the tone is not much stronger than the noise. (The outband ratio compares the sound amplitude contained in the tone to all other sound on the line.) Lowering the outband ratio may fix this problem — and create new ones. If the outband ratio is too ***low***, AccuCall might start confusing tones with voice.

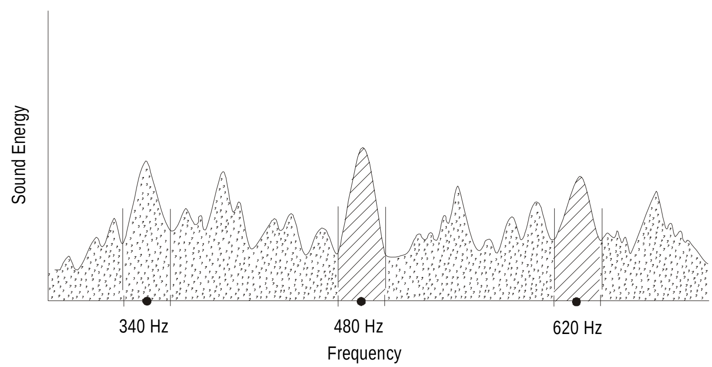


Figure 5. A Noisy Line

Figure 5 shows a noisy line, with a dual-frequency tone of 480 Hz and 620 Hz. In this situation, AccuCall would have trouble identifying the tone with the outband ratio test. In fact, in ***Figure 5*** the noise is bad enough that adjusting the outband ratio would not make a big difference in detection.

For more information on the various filter characteristics, see ***Understanding Filters on page 184***.

Testing a Tone

The Test New Tones function tests your PCPM table to see if it can make the driver recognize a real tone on a line. Do this after adding or editing a tone to ensure that your PCPM table is still functional.

➢ ***Do the following to test a tone:***

1. Go to the Tools menu and select the Test PCPM Table tab.

Test PCPM Table Dialog Box

The Test PCPM Table dialog box appears.

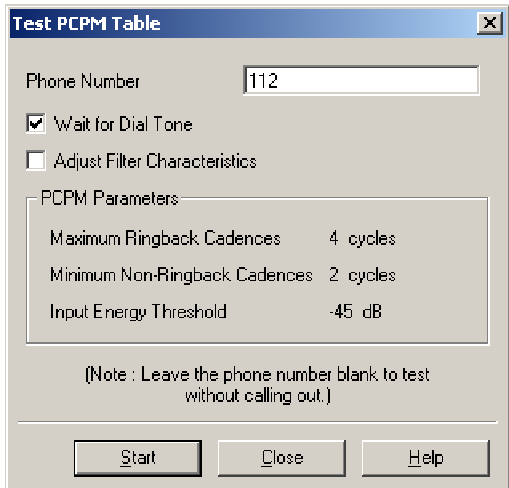
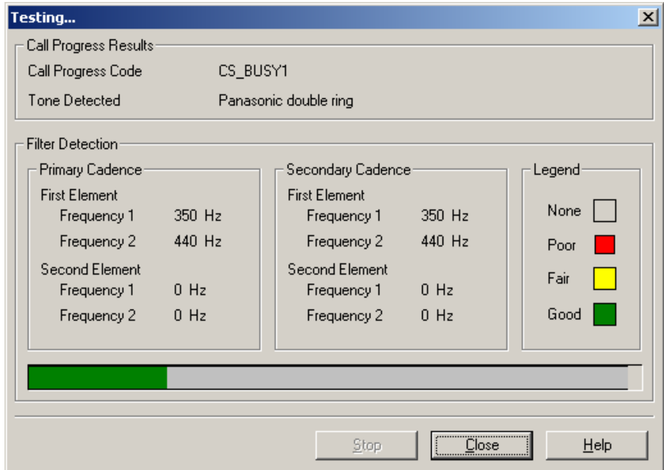


Figure 6. Setting Up the Test

2. Make sure that your board is properly connected to a line going to a CO or PBX. Use the Setup tab in the Options menu to select the channel that you want to use. Connect that channel to a phone line. If you check the Wait For Dial Tone box, the board does not call out until it senses that the channel is physically connected to a phone line.
3. Enter the phone number that you want to call and make sure that the line you are calling is generating the desired signal. In this example, we are calling extension #112 on a PBX, and extension #112 is currently off hook.
4. Check the Adjust Filter Characteristics box so that AccuCall can adjust frequency characteristics on the fly if it is having trouble learning the tone. Leave it unchecked if you want the filter characteristics to remain the same.

5. Click Start to begin the test.

Testing Dialog Box



The bar flashes on and off as AccuCall compares the PCPM table to the incoming tone. If the bar is green most of the time, AccuCall locks into the correct tone and displays the Call Progress Code, the name of the tone, and its frequencies. The table is functioning correctly. Click Close.

If the bar is red or yellow, AccuCall is having trouble. Let the test proceed for thirty seconds and then click Stop. If you did not check the Adjust Filter Characteristics box in step 4, do that now and try again. You can also adjust parameters by hand. See ***Understanding Filters on page 184*** and ***Start Parameters on page 190***.

If this does not help, the particular tone might not be correctly entered in your PCPM table, or the actual tone on the line might not be what you think it is. To check the second possibility, see ***Recording a Tone on page 218***.

Recording a Tone

The Record Tone function is useful for troubleshooting. You can use AccuCall to record a tone into the µlaw PCM format (.mlw) and then analyze the file yourself or send it to Brooktrout Technical Support.

➢ ***Do the following to record a tone:***

1. Go to the Tools menu and select the Record Tone command.

   The Record Tone dialog box appears.

2. Set up the call.

   Make sure that your board is properly connected to a line going to a CO or PBX. Use the Setup command in the Options menu to select the channel that you want to use. Connect that channel to a phone line. If you check the Wait For Dial Tone box, the board does not call out until it senses that the channel is physically connected to a phone line.

   Enter the phone number that you want to call and make sure that the line you are calling is generating the desired signal.

3. Choose a name for the file.

   Because the file is in the µlaw PCM format, give it a ***.MLW*** suffix.

4. Click Start.

   The Recording dialog box appears.

   A blue bar indicates the length of time you have been recording. Once you have enough data, click Stop.

   The file is saved in the same directory as AccuCall or where you specified when you entered a full path.

Improving Tone Detection

This section explains some of the problems that can arise when AccuCall tries to identify a tone. It also explains how to solve certain tone detection problems. Before reading further, learn how to use AccuCall and gain an understanding of the Filter, Start, Deglitching, and Answer parameters, described in this chapter.

Cadences

AccuCall can distinguish two tones with the same frequencies only by their cadences. Using similar cadences leads to poor tone detection.

Defining Cadence Overlap

Cadences can be divided into two categories: unique and non-unique.

Non-unique tone

A non-unique tone has a cadence that overlaps with the cadence of another non-unique tone. Overlapping means that one tone has one or more elements that fall within the tolerances of another tone's elements.

For example, consider the following:

	First Element	**Second Element**
Range	256–320 and 240–304	192–256 and 208–272
First Tone	288 ±32 ms	288 ±32 ms
Second Tone	272 ±32 ms	240 ±32 ms

The element durations of the two tones overlap. If the frequencies are close as well, AccuCall cannot distinguish the tones.

To be non-unique, both the First Element and the Second Element must fall within the range of another tone's First Element and Second Element. If one of the characteristics does not overlap or if the First Element overlaps with one tone and the Second Element overlaps with yet another tone, AccuCall can still distinguish the tones.

In a double cadence tone, the First Element and the Second Element must overlap in both the primary and the secondary cadence for the tone to be non-unique.

Unique tone

A unique tone has a cadence that does not overlap with the cadence of any other tone. AccuCall can easily detect unique tones.

Suppose that the second tone on page 2-43 has a First Element of 560 ms and a Second Element of 496 ms. Its First Element range is 480-640 ms and its Second Element range is 432–560 ms. These ranges do not overlap with the respective ranges of the first tone (256–320 and 192–256), so that AccuCall can distinguish the tones.

Demonstrating Cadence Overlap

If you have two non-unique tones, AccuCall might not be able to tell them apart. Consider the following examples.

Non-Overlapping Cadences

Table 12 shows the cadences of two single-frequency tones. Tone 1 is the Inter-Tel fast busy tone and Tone 2 is unspecified. The frequency for both is 487 Hz. The tones appear in the PCPM table in the order indicated.

Table 11. Non-Overlapping Cadences

	First Element	First Element Range	Second Element	Second Element Range
Tone 1	288 ms	256–320 (±11%)	224 ms	192–256 (±14%)
Tone 2	256 ms	240–272 (±6%)	272 ms	256–288 (±6%)

Figure 7 graphically illustrates the tone cadence ranges of the First Element and Second Element described in *Table 12*. A zero marker indicates the start of each element. For each tone in each element, two vertical markers define the minimum and maximum duration of the tone, and a crossed marker indicates the center of the range.

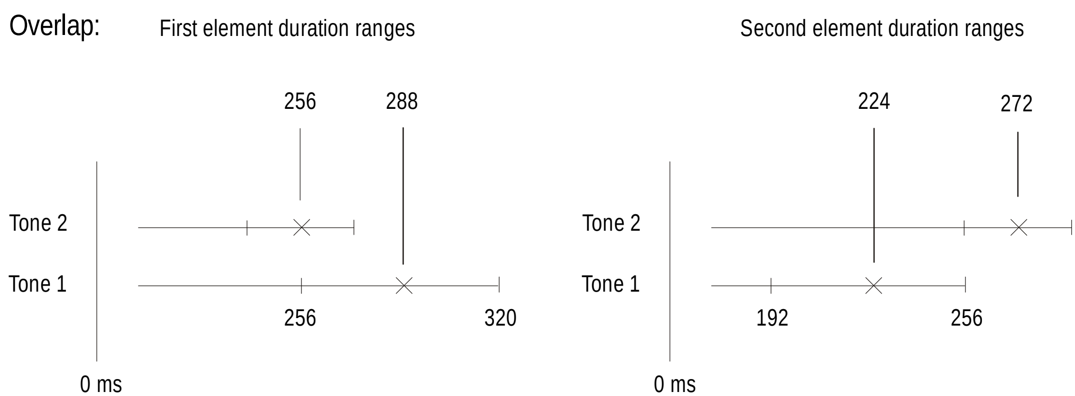


Figure 7. Non-Overlapping Cadences

When testing the PCPM table using a real Inter-Tel fast busy tone, AccuCall identifies the tone correctly as Tone 1 six out of six times. Only Tone 1 consistently matches the specified parameters. Tone 2 is not a candidate, because its range does not overlap with the range specified for Tone 1 in the Second Element. Although each tone has the same frequency, AccuCall successfully differentiates between the tones because each tone has a unique cadence.

Overlapping Cadences

In *Table 12 on page 222*, Tone 2 is defined with larger tolerances. Tone 1 (Inter-Tel fast busy) is unchanged. AccuCall cannot consistently differentiate these two tones, because the tones overlap in both elements. In an actual test, AccuCall identified an Inter-Tel fast busy tone as Tone 1 three times, and as Tone 2 three times.

Table 12. Overlapping Cadences

	First Element	First Element Range	Second Element	Second Element Range
Tone 1	288 ms	256–320 (±11%)	224 ms	192–256 (±14%)
Tone 2	256 ms	224–288 (±12%)	272 ms	240–304 (±12%)

Figure 8 graphically illustrates the tone cadence ranges of the first and second element described in *Table 12*.

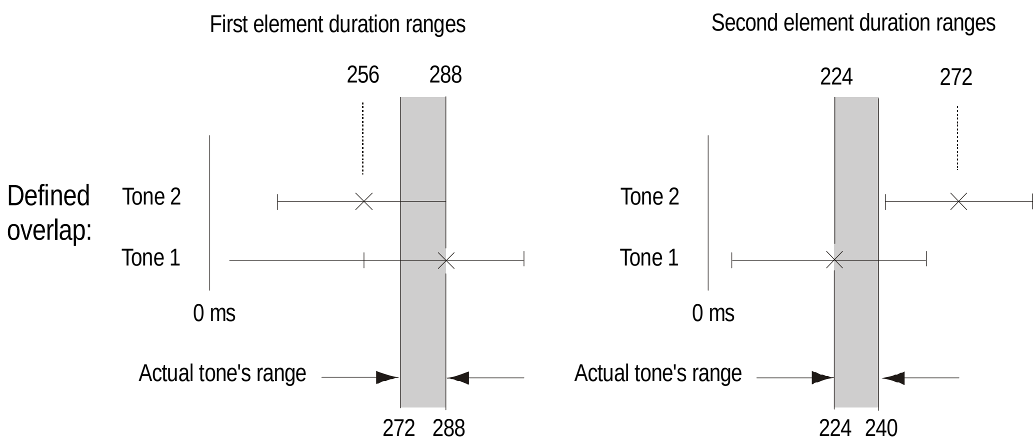


Figure 8. Overlapping Cadences

Detection is poor because sometimes AccuCall detects the Second Element of the Inter-Tel fast busy to be as low as 240 ms, and the First Element to be as high as 272 ms. When this happens, Tone 1 more closely matches the definition of Tone 2. The actual fast busy tone generated by the PBX might not be this erratic. AccuCall might be receiving bad samples or overestimating the First Element due to transition samples. For more information on transition samples, see *Comparing Tone Creation and Tone Learning on page 213*.

If you reverse the order of the tones in the PCPM table, AccuCall identifies the Inter-Tel fast busy as Tone 2 every time. AccuCall is more likely to identify whichever tone is listed first as the physical tone.

Fixing Cadence Overlap

Table 13 describes what the PCPM table looks like if you halve the Second Element range of Tone 1 and rename it Tone 1.1:

Table 13. Fixing Cadence Overlap

	First Element	First Element Range	Second Element	Second Element Range
Tone 1.1	288 ms	256–320 (±11%)	224 ms	208–240 (±7%)
Tone 2	256 ms	224–288 (±12%)	272 ms	240–304 (±12%)

In ***Table 13***, Tones 1.1 and 2 no longer overlap, but their ranges still include 240 ms. AccuCall might still identify the Inter-Tel fast busy as Tone 2, because sometimes it detects the tone's Second Element as 240 ms, and this is right on the line between Tone 1.1 and Tone 2.

In Summary: ***If the physical tone falls within the tolerances of a tone definition**, **AccuCall might identify it as that tone definition.***

Frequencies

Tones that are close in frequency can create problems in tone detection and filter assignment.

Defining Filter Overlap

Tone frequencies are divided into the following categories:

Identical frequency
An identical frequency is exactly the same as an assigned filter in the current filter table.

Non-unique frequency
A non-unique frequency is 1–19 Hz away from an assigned filter in the current filter table.

Too close frequency
A too-close frequency is 20–40 Hz away from an assigned filter in the current filter table.

Unique frequency
A unique frequency is 41 or more Hz away from an assigned filter in the current filter table.

For example, assume you have a blank PCPM table with one filter set to 360 Hz.

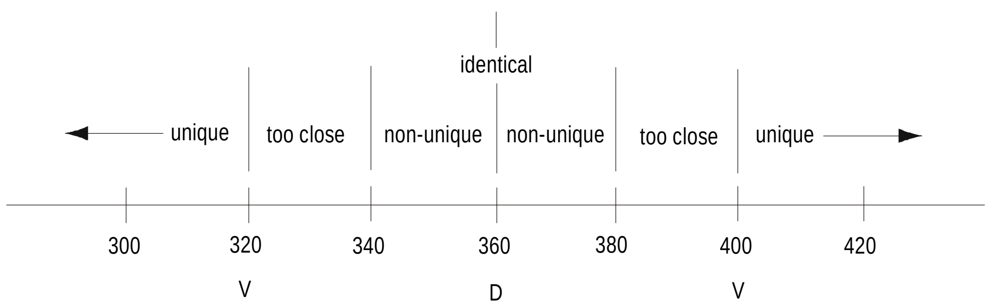


Figure 9. Tone Frequency Categories

Figure 9 is centered on the open filter at 360 Hz. The four possible frequency categories are shown arranged around the open filter. ‘D’ marks the location of the defined filter, and ‘V’ marks the closest valid frequencies where you could assign a filter.

If you use the Learn New Tones function with the Align Frequencies option enabled, the following occurs:

- A non-unique frequency (341–359 Hz and 361–379 Hz) aligns with the center frequency, 360 Hz.
- A unique frequency (less than 320 Hz and greater than 400 Hz) causes an undefined frequency to appear in the Filters tab. To load the PCPM table to the driver and test it, assign a filter to the unique frequency.
- A too-close frequency (321–340 Hz and 380–399 Hz) also causes an undefined frequency to appear in the Filters tab. However, AccuCall does not let you assign it a filter at its exact frequency, since the new filter would overlap with the filter at 360 Hz.

To solve the too-close frequency problem, assign a filter that is close to the new frequency but is not within 40 Hz of the 360 Hz filter. For example, if the too-close frequency is 330 Hz, assign a new filter at 320 Hz, then manually align the too-close frequency tone with the new filter assignment.

If a filter within 80 Hz of the 360 Hz filter already exists, this solution does not work, because you cannot assign a filter between the two existing ones. Instead, align the too-close frequency to the closest open filter. For example, if the too-close frequency is 330 Hz, and there are open filters at 300 Hz and 370 Hz, align the tone with the 300 Hz filter. Also try reassigning filters so that they lie between two-tone frequencies, rather than right on top of one and far away from another. This could improve overall detection, although you then must edit ***all*** the affected tones, rather than just the one you were trying to add.

Alternatively, you can edit the problem tone and check the Cadence Only box. AccuCall then ignores the frequency when it tries to identify that particular tone.

In a dual-frequency tone, the component frequencies can be too close, unique, non-unique, or any combination. The important thing is not the status of the component frequencies, but whether the overall tone is unique.

The overall dual-frequency tone is unique if its component frequencies fall within filters that no other dualtone uses in the same combination. For example: you have a PCPM table with open filters at 300 Hz, 360 Hz, and 420 Hz. There are three dual-frequency tones in the table. They have frequencies in Hz of 300 and 360, 300 and 420, and 360 and 420. Since the three tones fall into different filter slots, AccuCall can distinguish them all. If you added a fourth tone of 365 Hz and 401 Hz, the third and fourth tones would both use the filters at 360 and 420 Hz. AccuCall must distinguish these tones using their cadences.

Defining Identical Frequencies

Tones from the same switch often have the same frequencies but different cadences. You can define these tones by just specifying the cadence in the Edit Tones tab and checking the Cadence Only box.

A better way to define these tones is to also specify the frequency. In the future you might add tones with different frequencies than your current tones, but similar cadences. To specify frequency you must uncheck the Cadence Only box and update the Filter Table.

Aligning Frequencies Automatically

When you check the Align Frequency box in the Learn New Tones dialog box, AccuCall changes the value of a newly learned tone's frequency to match the closest assigned filter. However, without the right filter characteristic settings, you might end up aligning to a filter that is assigned incorrectly. The following example describes how to use the Align Frequency option.

Assume you have a PCPM table with a Bell network dualtone with frequencies of 440 and 480 Hz. When you added the Bell network tone, you selected the Adjust Filter Characteristics option (see ***Figure 6 on page 216***) to ensure optimal filter characteristic settings.

Now you want to add Panasonic 308 dualtones with frequencies of 341 and 445 Hz. If you check the Align Frequency box, AccuCall aligns the Panasonic frequency of 445 Hz with the existing filter of 440 Hz.

To see if AccuCall can identify the Panasonic tone, test it while watching the detection bar. If detection is Fair or Poor, try reversing the order of the tones in the PCPM table.

➢ ***Do the following to test the tone:***

1. Clear the tones from the PCPM table.
2. Learn the Panasonic tone.
3. Test it using Adjust Filter Characteristics.
4. Add the Bell tone and verify it without using Adjust Filter Characteristics.

In general, check the Adjust Filter Characteristics box only when you are adding the first and most difficult tone to the PCPM table. If you are having trouble learning a tone, it can help to clear the PCPM table and try learning that tone first using the Adjust Filter Characteristics option.

You can also experiment with the value of the aligning filter. In this example it was 440 Hz, but you might get better detection with a value between 440 and 445 Hz. Finally, if the filter characteristics change, record them for future reference.

Optimizing Tone Detection Using Quick Count

A single tone is unique if it occupies its own filter. A dualtone is unique if no other dualtone uses the same two filters. You can optimize detection of these tones by experimenting with the values of the aligning filters. Another way to optimize tone detection is to select the Quick Count option, which detects tones based only on frequency. Activate this option by setting the Quick Count parameter in the Edit Tones dialog box. You must also check the Quick Scan box when testing a tone.

➢ ***When using* Quick Count*, the following procedure generally applies:***

1. Use the Learn New Tone function to analyze the tone.
2. In the Edit Tones dialog box, set Quick Count to about one quarter of the First Element Duration or 48 ms, whichever is greater. The tone's Durations become equal to Quick Count and grey out.
3. Select the tone Type to be Non-Ringback.

There are two categories of unique tones: burst tones and regular tones. Quick Count can be useful in both situations: in the first, where cadence is irrelevant, and in the second, to speed up detection.

Burst Tones

Burst tones either have an irregular cadence or no cadence at all. Examples include fax and pager tones. Quick Count helps optimize detection.

Use Quick Count if the frequency is non-unique, but only in the case of continuous tones (such as a dial tone) or in cases where the First Element of the tone is longer than any other tone.

For examples of dealing with burst tones, see ***Detecting Fax Tones on page 232***, or ***Detecting Pager Tones on page 233***.

Regular Tones

Regular tones have frequencies ***and*** regular cadences. Ordinarily, AccuCall uses both to detect the tone. Unfortunately, it has to wait and sample over many cycles before it can identify the cadence. This means detection can take well over a second. Quick Count can speed detection up dramatically because it samples the frequency only for a brief time. However, if your tones do not use unique filter assignments, Quick Count can cause poor detection.

Tone Detection

Table 14 shows the preferred method for defining each type of tone and offers an alternative method.

Table 14. Summary of Tone Detection Methods

Tone Type		Method	Alternative
Frequency	**Cadence**		
Identical	Unique	Cadence + Frequency (recommended)	Cadence Only
Non-unique	Unique	Cadence + Frequency & Align Frequency (recommended)	Cadence Only
Non-unique	No Cadence	Quick Count	---
Too Close	Non-unique	Manual Align & Quick Count (only if Align yields unique frequencies)	Manual Align & Cadence + Frequency (slower detection)
Too Close	Unique	Manual Align & Quick Count (only if Align yields unique frequencies)	Manual Align & Cadence + Frequency or Cadence Only (slower detection for both)
Unique	Non-unique	Quick Count	Cadence + Frequency (slower detection)
Unique	Unique	Quick Count	Cadence + Frequency or Cadence Only (slower detection for both)
Unique	No Cadence	Quick Count	---
Multiple	Unique	Cadence Only (see page 2-60)	---
Multiple	Non-unique	---	---

---	No method available
No Cadence	Burst Tone (cadence does not exist or is irregular)
Multiple	Multiple-frequency tone (tone has three or more frequencies)
Cadence Only	Check the Cadence Only box in the Edit Tones dialog box
Quick Count	Set the Quick Count parameter in the Edit Tones dialog box
Align Frequency	Check the Align Frequency box in the Learn New Tones dialog box
Manual Align	Align the frequency with a valid filter using the Edit Tones dialog box

Troubleshooting

The following is a list of commonly encountered problems with tone detection. Before reading this section, familiarize yourself with the information in the ***Cadences on page 219*** and ***Frequencies on page 224***. Keep in mind that the solutions given in this section are for only the most common problems, and since every situation is unique, these solutions might not apply.

Detecting a Problem Tone

Problem

➢ ***"I can't detect a problem tone".***

Solution

Start with a blank PCPM table. Go to the Learn New Tone dialog box and set the Minimum Learning Cycles field to 15.

After learning the problem tone, try detecting it using the Test PCPM Table function. Mark the Adjust Filter Characteristics checkbox before you begin. If enabling AFC detects the tone, disable AFC and add the other tones to your table.

In general, the tone that was most difficult to learn should define the filter characteristics of your PCPM table.

Overlapping Filters

Problem

- ***"How can I detect a tone that has a frequency within ±20 Hz of another tone?"***

Solution

1. Set the Filter Characteristics to their default values.
2. Learn New Tone function on each of the tones ***separately***, clearing the PCPM table before each time. Record the cadences, frequencies and filter characteristics of both tones.
3. Verify the tones separately.

   If the filter detection of one of the tones is fair or poor, try again with the Adjust Filter Characteristics option checked. If both tones are fair or poor, check the Adjust Filter Characteristics box and test the tone with the worst tone detection first. Whenever the Filter Characteristics change, record the new values.

   As the Filter Characteristics change, the progress bar should improve with time. If the First Element Duration is very short, increase Maximum Ringback Cadences and Minimum Non-Ringback Cadences.

   Repeat the tests independently, with the Adjust Filter Characteristics box unchecked.
4. Add the second tone by aligning it to the first.

   Choose the tone that gave you the most trouble during testing and start with it in your PCPM table. Use the Learn New Tone function to add the second tone to your PCPM table, this time selecting the Align Frequency option.

   AccuCall assigns the second tone the same frequencies as the first tone. This does not affect the filter characteristics. AccuCall adjusts them only when it is testing a tone.
5. Test the first tone.

   Test it with and without the Adjust Filter Characteristic box checked.
6. Test the second tone.

   AccuCall should detect both tones successfully. If detection is Fair or Poor, do not try to detect again using Adjust Filter Characteristics. Go back to step 3, this time swapping the roles of the first tone and the second tone.

   Also experiment with the value of the aligning filter. Use values that fall between the frequencies of both tones. Make sure you update both the filter and PCPM tables.

Detecting Fax Tones

Problem

> ***"How can I detect an incoming fax call or fax answer tone?"***

Solution

Go to the Edit Tones dialog box and enter the values given below. Items in parenthesis refer to fax answer tones.

Name	Incoming Fax Tone (Fax Answer Tone)
Call Progress Code	CS_CNGDETCT or CS_SPECIALCP
Terminating	Checked
Cadence	Unchecked
Quick Count	64
First Element	
Frequency 1	1100 (2100)
Frequency 2	0
Duration	Not applicable
Second Element	
Frequency 1 & 2	Set both to 0.
Duration	Not applicable

Add the tone to your PCPM table. Go to the Filters tab and enter 1100 (2100) in a real-time filter slot.

When your application receives Call Progress Code CS_CNGDETCT or CS_SPECIALCP, branch your application to special logic that handles fax tones.

Detecting Pager Tones

Problem

➢ ***"How can I detect pager tones?"***

Solution

A paging service usually answers with a few short beeps and no ring. However, some have a single ring cycle. The frequencies of these beeps vary, but they are usually unlike any of the frequencies used in regular call progress tones.

Since the cadence is irregular, AccuCall should detect these tones using Quick Count. Unfortunately, AccuCall can have trouble analyzing the short beeps.

Go to the Edit Tones dialog box and enter the values given below.

Name	Pager Tone
Terminating	Checked or unchecked, depending on the tone's purpose
Cadence Only	Unchecked
Quick Count	64
First Element	
Frequency 1	As determined by the learning process
Frequency 2	As determined by the learning process (most pager tones are single-frequency)
Duration	Not applicable
Second Element	
Frequency 1 & 2	Set both to 0
Duration	Not applicable

Use the Learn New Tones function to determine the frequency of the pager tone. Go to the Filters tab and add the frequency to the appropriate real-time or non-real-time filter slot. Load the table to the driver and test it.

Dealing With 'Too-Close' Frequencies

Problem

➤ ***"How can I detect a tone with a frequency that is positioned too far to be aligned with an existing filter, but is still positioned within 40 Hz of that filter?"***

Solution

The frequency is a 'too-close' frequency (see ***Defining Filter Overlap on page 224***). For simplicity, assume that no filters are within 80 Hz of the existing filter.

Use the Learn New Tone function to characterize the tone. After finishing the analysis, enter the following values in Edit Tones:

Name

Too Close Tone

Terminating

This is checked or unchecked, depending on the tone's purpose.

Cadence Only

Unchecked

Quick Count

64

First Element Frequency 1 & 2

Change the frequencies to the values of the nearest valid, unassigned filters. See ***Defining Filter Overlap on page 224*** for an example of how to select a nearby, unassigned filter slot when you have a too-close frequency.

Duration

Not applicable

Second Element Frequency 1 & 2

Set both to 0.

Duration

Not applicable

Use the Learn New Tones function to determine the frequency of the tone. Go to the Filters tab and add the frequency to the appropriate real-time or non-real-time filter slot. Load the table to the driver and test it.

Whether you align a single or dual-frequency tone, you must not assign multiple tones to the same filter slots and still use Quick Count. AccuCall cannot distinguish such tones without cadence information.

If you have a dual-frequency tone with frequencies that are close to each other (say a Bell network tone of 440 and 480 Hz), filter constraints might force you to list it as a single-frequency tone. This is acceptable as long as the tone is unique in some way, either in filter assignment or in cadence.

Detecting Multiple-Frequency Tones

Problem

➢ ***"How can I detect complex tones that consist of more than two frequencies?"***

Solution

Use the Learn New Tone function to characterize the tone. Since the tone has three or more frequencies, it must be identified by its cadence only. Enabling the Cadence Only option tells the Driver to disable frequency analysis and detect on the basis of cadence only. If you cannot make the cadence unique, you are not able to detect the tone.

After you finish, go to the Edit Tone dialog box and make the following adjustments.

Name	Multiple Frequency Tone
Terminating	Unchecked
Cadence Only	Checked
Quick Count	–1
First Element Frequency 1 & 2	Set both to 0
Duration	As determined by the learning process
Second Element Frequency 1 & 2	Set both to 0
Duration	As determined by the learning process

When you are done, check to see if the analysis produced any undefined frequencies. Do not add these frequencies to the tone or to the filter table.

Checking the Terminating box would not make the tone terminating. AccuCall distinguishes terminating tones from non-terminating tones using frequency information.

9 - Using the Dialogic® Brooktrout® TECUpdate Utility

This chapter describes how one can use the Dialogic® Brooktrout® Technology Expansion Capability (TECUpdate) utility to display and upgrade the configured feature set on an installed board.

This chapter includes the following:

To successfully perform an upgrade, you need to know the serial number of your board and have an appropriate license key provided by Dialogic.

Use the Dialogic® Brooktrout® TECUpdate utility to record the serial number of your board before you contact Dialogic to order the license file. When you receive an email with your updated license file, you can again use the TECUpdate utility to download the license file and expand the functionality of your board.

Starting Dialogic® Brooktrout® TECUpdate

The Dialogic® Brooktrout® TECUpdate utility is installed as a separate software package. For more information:

- See the chapter on packaging your application in the ***Developer Guide***.
- Contact your authorized Dialogic reseller or application partner.

Distribute the TECUpdate utility to your end users even if you do not plan to distribute the Dialogic® Brooktrout® Configuration Tool, so that your customers can update their systems as needed.

The TECUpdate utility assumes that the:

- Firmware is downloaded to the board.
- Dialogic® Brooktrout® BOSTON driver is started.

You can run the TECUpdate utility from the file system (for example, Windows® Explorer), Start Menu or the command line:

- File System:

  `[`*`InstallDIR`*`]\TECUpdate.exe`

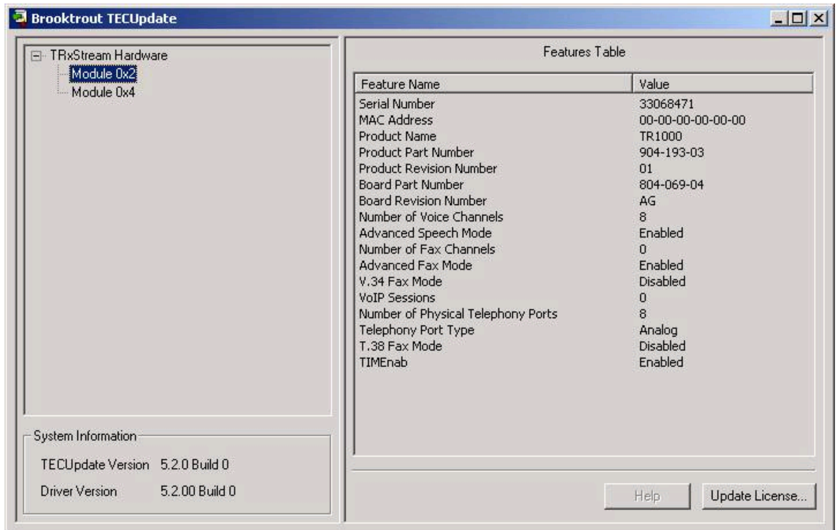
  where: `InstallDIR` is the directory where you installed the utility.
- Start Menu:

  Start → Programs → Brooktrout → TECUpdate_setup.exe
- Command Line:

  `c:\Brooktrout\boston\TECUpdate.exe`

The TECUpdate utility queries and displays information for the selected module. TECUpdate operates with Dialogic® Brooktrout® drivers from SDK 2.1.0 and above.

Following a successful start, the Brooktrout TECUpdate window appears. See ***Running the Dialogic® Brooktrout® TECUpdate Utility on page 241*** for specific steps to obtain module information.



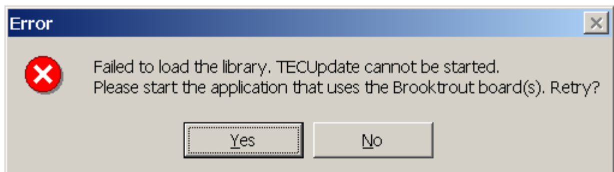
When Errors Occur

If an error occurs, it is recorded in the error log file (see ***Error Logging on page 243***). Contact Dialogic Technical Support with the error log information if you cannot resolve the problem.

If there are firmware or driver errors, the following Error dialogs are possible:

Driver Error

- Click Yes to retry after loading the driver. Before trying to run TECUpdate again, an Error dialog asks you to load the driver:



Yes

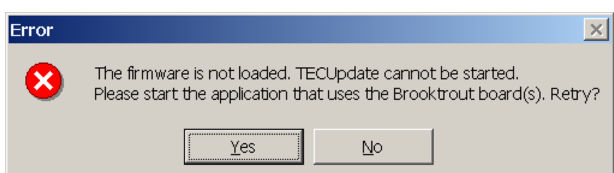
Tries to determine if the driver is started, then proceeds with the TECUpdate initialization process. If an error occurs, the dialog reappears.

No

TECUpdate utility exits.

Firmware Error

- Click Yes to retry after downloading the firmware. Before trying to run TECUpdate again, an Error dialog asks you to load the driver:



Yes

Tries to determine if the firmware is loaded, then proceeds with the TECUpdate initialization process. If an error occurs, the dialog reappears.

No

TECUpdate utility exits.

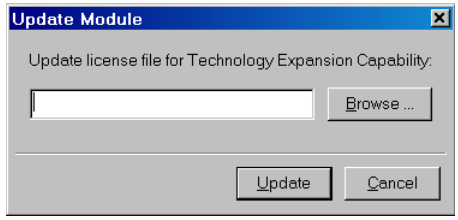
Running the Dialogic® Brooktrout® TECUpdate Utility

➢ ***To display and update information for each module in your system, do the following:***

1. Click on a specific module to display its information:
   - The left panel contains the tree for TRxStream Hardware. All modules installed in your system are shown.
   - The lower left panel contains the TECUpdate Version and Driver Version information.
   - The right panel displays the feature set for the selected module.
   - The lower right panel contains an online Help and Update License buttons. Update License allows you to update the feature configuration for a selected module using a specified license file.

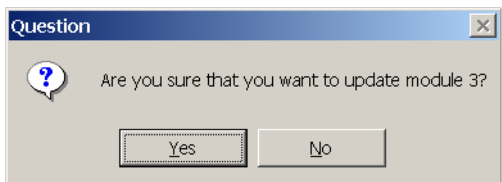
2. Click Update License to load the license file updates.

   The Update Module dialog appears:

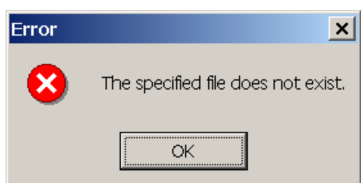


3. Click Browse to locate the license file (***.lic**), then click Update to begin the updating process.
4. Click Yes to begin the download of the license file.

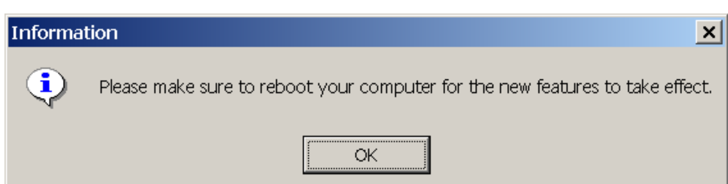
   A Question confirmation dialog appears before the update continues.



Click OK when an Error dialog appears to return to the Browse process (Step 3) and locate the correct file.



5. Click OK if boards were updated with the new license information. The Feature Table on the right panel reflects the most current information about the module.



6. Reboot your computer.

Error Logging

If TECUpdate encounters an error, an Error dialog box is displayed. Please see the error log file for more information. The error log file includes:

- TECUpdate version
- Driver version
- Diagnostic information

Provide the log file to Dialogic Technical Support for more information on solving this error. The ASCII text file is called ***TECUpdate.log*** and is stored in the current application working directory. Subsequent errors are appended to this log file.

10 - Downloading Firmware

This chapter describes how one can download the firmware to the board.

The chapter includes the following:

Introduction

To install the software, you can download the firmware files and optionally configure call control.

For Linux and Solaris operating systems, the procedures for downloading firmware are described here.

If your operating system is Windows, you can use the Dialogic® Brooktrout® Configuration Tool to download the firmware and configure the call control parameters.

To configure call control, see ***Using a Call Control Configuration File on page 145***) and ***Volume 6***, ***Appendix A*** of the ***Bfv API Reference Manual.***

Firmware Files

Table 15 lists the firmware files. See the Release Notes that came with your software CD for further information.

Table 15. Firmware and Flash Update Files

File	Use
bostvb.dll	Firmware for the SR140 virtual module
cp.bin	Control processor (PPC) firmware for all platforms
dsp1000.hex	High density DSP firmware for TR1000 voice and fax modules without V.34 fax (includes V.17 fax)
dsp1000_ld.hex	Low density DSP firmware for TR1034 analog and BRI modules or TruFax® analog and BRI boards (includes V.17 and V.34 fax)
dsp1000_ud.hex	Ultra high density DSP firmware for multi-span TR1000 voice and fax modules (non-V34)
dsp1000_v34.hex	High density DSP firmware for TR1034 voice and fax with V.34 fax and T.38 fax
dsp1034_ud.hex	Ultra high density DSP firmware for multi-span TR1034 voice and fax modules with V.34 fax and T.38 fax

Updating Boot ROM Flash

Typically, it is not necessary to update your Boot ROM. Use the following procedure to update your Boot ROM Flash firmware only when told to do so by a Dialogic Technical Support Representative.

The *boston* driver must be running when you update the boot ROM flash.

➢ ***Use the following procedure to update your Boot ROM Flash firmware.***

1. Ensure that each board has a unique module number. Maintain a current list. See ***Determining the Board Module Number on page 2*** for more information about modules and module number assignment. You can obtain a list of modules using the ***modinfo*** program.
2. Use the ***firm*** program to perform a flash update (do not interrupt the update process):

```
firm -m <module> -t 2 flashupd.bin
```

```
where: <module> is the hardware module number.
```

Note: This process takes less than a minute to complete. It is important not to interrupt the update process. If you interrupt the update process, you cannot reboot your board.

Manually Downloading the Firmware

You must download the firmware each time you boot the machine or restart the device driver.

To download the firmware to a module, you must run the ***firmload*** or ***firm*** program, both of which are located in the ***brooktrout/boston/bfv .api/<platform>/bapp.src*** directory.

The ***firmload*** program loads the control processor and DSP firmware on all Dialogic® Brooktrout® modules found in the system.

The `-c` option for the ***firm*** and ***firmload*** commands or the ***config_spec_value*** field of ***BfvFirmwareDownload*** sets the actual number of channels supported on the module.

First, start the driver if it is not already started, and if your operating system platform requires it.

Then, change to the ***<install_root>/bfv.api/<platform>/bapp.src directory*** and run the ***firmload*** or ***firm*** program.

To run the ***firmload*** program:

```
firmload -c num_channels brooktrout/boston/fw
```

Where `num_channels` is the number of channels configured to receive a firmware download.

For example:

```
firmload -c 48 brooktrout/boston/fw
```

The firmware consists of two types, by number, which must be downloaded in the proper sequence. The ***firmload*** program automatically takes care of identifying the proper files and downloading them in the correct sequence. If you wish to download the firmware using the ***firm*** program, you must download each module separately and specify the number of channels (`-c`) in the first ***firm*** command:

```
firm -m <module> -c <num_channels> -t <firmtype>
<firmname>
```

➢ ***The sequence is as follows:***

1. Type 2 (PROC_APP) = Control processor firmware (`cp.bin`)
2. Type 1 (DSP_APP) = DSP firmware, use one of the following DSP firmware files:

`bostvb.dll`	SR140 host based fax
`dsp1000.hex`	TR1000 boards in T1 robbed-bit, T1 or E1 ISDN
`dsp1000_ld.hex`	TR1000 Series analog and BRI boards and TruFax® analog and BRI boards
`dsp1000_ud.hex`	Multi-span TR1000 Series ISDN boards (non-V.34)
`dsp1000_v34.hex`	TR1034 boards with V.34 fax and T.38 fax
`dsp1034_ud.hex`	Multi-span TR1034 boards with V.34 fax and T.38 fax

For example:

```
firm -m 2 -c 30 -t 2 cp.bin
firm -m 2 -t 1 dsp1000.hex
```

It takes about 10 seconds per module for the firmware to download and for the module to begin operating. Proper operation is indicated by the module status LED slowly flashing green.

See the descriptions of these programs in the developer guide chapter about sample applications and utilities.

When the firmware is downloaded to a module for the first time, the assigned ordinal channel numbers start wherever the assignment left off on the previous module. As the system initializes the modules, this numbering process creates a continuous ordering of the channel assignments across all the modules in the system. On later downloads, each module’s ordinals begin at the same location, regardless of any decrease or increase in the channel count of a lower-numbered module. Therefore, if you decrease the channel count for a lower numbered module, the process creates gaps in the channel numbering assignments, possibly affecting your application. If you attempt to increase the channel count above any module’s initial channel count, the system ignores the added channels.

➢ ***For the following situations, restart the driver whenever you want to:***

1. Get a continuous assignment of channel numbers after decreasing the channel count on any module.
2. Increase the number of channels above a module's initial channel count.

Note: You must reboot the system to restart the driver.

Appendix A
Compliance Information

This appendix describes how the Dialogic® Brooktrout® boards conform to compliance standards.

It has the following sections:

- *Electromagnetic Compatibility Statements on page 254*
- *Telecommunications Compliance Statements on page 259*
- *Safety Compliance Statements on page 266*

Note to developers and system integrators: The following compliance information and statements must be provided to your customer or end user as part of your system documentation.

Electromagnetic Compatibility Statements

Electromagnetic compliance requirements include country specific statements for:

- ***United States of America on page 255***
- ***Canada on page 257***
- ***European Union on page 257***
- ***Japan on page 258***

The Federal Communications Commission (FCC) in the United States and Industry Canada (IC) in Canada regulate all electronic devices that connect to the telephone system and/or generate radio frequency signals. Additionally, all computing devices utilizing clock frequencies in excess of 10 kHz must be tested for compliance with RF emission limits set by the FCC and IC. The Dialogic® Brooktrout® board is such a device and must comply with the Class A or Class B regulations as specified below.

The following statements should be conspicuously located in bold letters in the end user system documentation.

United States of America

Mandatory Statements

Changes or modifications to this unit not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

You must also add the following paragraph if shielded cables were used when testing your product:

> In order to maintain compliance with FCC limits, shielded cables must be used with this equipment. Operation with non-approved equipment or unshielded cables is likely to result in interference to radio & television reception.

Class A Statement (for "Class A" products)

NOTE: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Class B Statements (for “Class B” products)

NOTE: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this is equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and the receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

You might also find the following booklet helpful, prepared by the FCC: “***How to Identify and Resolve Radio-TV Interference Problems***.” This booklet is available from the U.S. Government Printing Office, Washington, D.C. 20402

Canada

Class A Statements (for “Class A” products)

This Class A digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la class A est conforme à la norme NMB-003 du Canada.

Class B Statements (for “Class B” products)

This Class B digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la class B est conforme à la norme NMB-003 du Canada.

European Union

Class A Statements (for “Class A” products)

This equipment complies with the requirements of CISPR 22 (EN 55022) for Class A Information Technology Equipment (ITE).

Warning: This is a Class A product. In a domestic environment this product may cause radio interference in which case the user is required to take adequate measures.

Class B Statements (for “Class B” products)

This equipment complies with the requirements of CISPR 22 (EN 55022) for Class B Information Technology Equipment (ITE).

Japan

VCCI Class Statements

Class A ITE

Class A Statement (For Class A Products).

この装置は、情報処理装置等電波障害自主規制協議会（ＶＣＣＩ）の基準に基づくクラスＡ情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

Translation:

This is a Class A product based on the standard of the Voluntary Control Council for Interference by Information Technology Equipment (VCCI). If this equipment is used in a domestic environment, radio disturbance may occur, in which case, the user may be required to take corrective actions.

Class B ITE

Class B Statement (For Class B Products).

この装置は、情報処理装置等電波障害自主規制協議会（ＶＣＣＩ）の基準に基づくクラスＢ情報技術装置です。この装置は、家庭環境で使用することを目的としていますが、この装置がラジオやテレビジョン受信機に近接して使用されると、受信障害を引き起こすことがあります。

取扱説明書に従って正しい取り扱いをして下さい。

Translation:

This is a Class B product based on the standard of the Voluntary Control Council for Interference from Information Technology Equipment (VCCI). If this is used near a radio or television receiver in a domestic environment, it may cause radio interference. Install and use the equipment according to the instruction manual.

Telecommunications Compliance Statements

The following customer information must be provided to customers with each unit of approved terminal equipment.

The statements provided below should be conspicuously located in the end user system documentation:

United States of America

Equipment Approved After July 23, 2001

This equipment complies with Part 68 of the FCC rules and the requirements adopted by the ACTA. On the circuit side (bottom side) of this equipment is a label that contains, among other information, a product identifier in the format USAAAEQ##TXXXX. If requested, this information must be provided to the telephone company.

Ringer Equivalence Number (REN)

The REN is used to determine the number of devices that might be connected to a telephone line. Excessive RENs on a telephone line can result in the devices not ringing in response to an incoming call. In most but not all areas, the sum of RENs should not exceed five (5.0). To be certain of the number of devices that can be connected to a line, as determined by the total RENs, contact the local telephone company.

For products approved after July 23, 2001, the REN for this product is part of the product identifier that has the format US:AAAEQ##TXXXX. The digits represented by ## are the REN without a decimal point (e.g., 03 is a REN of 0.3). For earlier products, the REN is separately shown on the label.

Connection to Telecommunications Networks

A plug and jack used to connect this equipment to the premises wiring and telephone network must comply with the applicable FCC Part 68 rules and requirements adopted by the ACTA. A compliant telephone cord and modular plug is provided with this product. It is designed to be connected to a compatible modular jack that is also compliant. Please refer to the installation instructions provided with this equipment for details.

Please refer to the installation instructions provided with this equipment for details concerning the Universal Service Order Codes (“USOC”) that are applicable to this equipment.

If this equipment causes harm to the telephone network, the telephone company will notify you in advance that temporary discontinuance of service might be required. But if advance notice isn't practical, the telephone company will notify the customer as soon as possible. Also, you will be advised of your right to file a complaint with the FCC if you believe it is necessary.

The telephone company can make changes in its facilities, equipment, operations or procedures that could affect the operation of the equipment. If this happens the telephone company will provide advance notice in order for you to make necessary modifications to maintain uninterrupted service.

There are no user replaceable or serviceable components in this equipment. If trouble is experienced with this equipment, for repair or warranty information, please refer to the appropriate section of the general customer instructions for this equipment. If the equipment is causing harm to the telephone network, the telephone company can request that you disconnect the equipment until the problem is resolved.

This equipment must not be used on party lines. Connection to party line service is subject to state tariffs. Contact the state public utility commission, public service commission or corporation commission for further information.

FCC Rules Regarding FAX Branding

The Telephone Consumer Protection Act of 1991 makes it unlawful for any person to use a computer or other electronic device, including FAX machines, to send any message unless such message clearly contains in a margin at the top or bottom of each transmitted page or on the first page of the transmission, the date and time it is sent and an identification of the business or other entity, or other individual sending the message and the telephone number of the sending machine or such business, other entity, or individual. (The telephone number provided cannot be a 900 number or any other number for which charges exceed local or long-distance transmission charges.)

To program this information into your FAX machine, you should complete the following steps:

Users: To program this information into your fax machine, follow the procedure described in your user manual.

Developers: You must include facilities in your application to enable the user to enter the required information. Use the ***BfvFaxHeader*** function with the Dialogic® Brooktrout® board to place this information on each transmitted page as required. You must also include, in your user manual, instructions for entering this information into your system.

Note to OEM Partners (final equipment assemblers):

For information related to power supply source requirements, signal levels, etc., please refer to the installation instructions and general specifications provided with this equipment.

The mounting of the approved unit in the final assembly must be made so that:

- The approved unit is isolated from exposure to any hazardous voltages within the assembly.
- Adequate separation and restraint of cables and cords must be provided.
- The circuitry from the approved unit to the telephone line must be provided in wiring that carries no other circuitry (such as PC or PR leads) unless specifically allowed by the rules.
- PC board traces carrying tip and ring leads shall have sufficient spacing to avoid surge breakdown.
- The final assembler shall provide in the consumer instructions all applicable customer information.

- The approval label shall be placed on the exterior of the cabinet for each type of approved device contained therein if the approved device is enclosed in an assembly, and not readily accessible.
- Modular plugs or jacks shall be provided which comply with TIA-968-A Chapter 6 requirements for dimensions, tolerances and metallic plating.

FCC Regulations For Connecting to a T1 Interface

The Federal Communications Commission (FCC) has established rules that permit a Dialogic® Brooktrout® board to be directly connected to the telephone network:

Standardized jacks are used for connections.

This equipment can not be used on coin service provided by the telephone company. Connection to party lines is subject to state tariffs. (Contact your state public utility commission or corporation commission for information).

A malfunctioning circuit can harm the telephone network.

Disconnect a malfunctioning Dialogic® Brooktrout® board from the telephone network until you determine the cause of the malfunction and repair it. If a malfunctioning Dialogic® Brooktrout® board remains connected, the telephone company can temporarily disconnect service.

The Dialogic® Brooktrout® board is approved as a DSX-1 device. Federal regulations (FCC Part 68) prohibit connection of a DSX-1 device to the network without an FCC approved Channel Service Unit (CSU). Customers connecting this device to the network shall, upon request of the telephone company, inform the telephone company of the particular lines to which such connections are made and the FCC registration of the protection device (CSU).

The CSU has been designed to prevent harm to the T1 network. If the telephone company finds that the equipment is exceeding tolerable parameters, the telephone company can temporarily disconnect service, although they will attempt to give you advance notice if possible.

If the telephone company alters their equipment in a manner that will affect use of this device, they must give you advance warning so as to give you the opportunity for uninterrupted service. You will be advised of your right to file a complaint with the FCC.

Under the FCC rules, no customer is authorized to repair this equipment. This restriction applies regardless of whether the equipment is in or out of warranty.

Before connecting the Dialogic® Brooktrout® board to telephone service, you must give a representative of the local telephone company the following information:

- The telephone numbers (Port ID) to which the Dialogic® Brooktrout® board is connected.
- SOC: 6.0P HP02
  6.0Y HP03
- FIC: 04DU9-BN 1.544 Mbps SF
  04DU9-DN 1.544 Mbps SF+B8ZS
  04DU9-1KN 1.544 Mbps ESF
  04DU9-1SN 1.544 Mbps ESF+B8ZS
- The type of wall jack required: USOC-RJ-48C
- The FCC Registration number: Labeled on back of board
- FCC registration of the protection device (CSU)

Canada

NOTICE: This equipment meets the applicable Industry Canada Terminal Equipment Technical Specifications. This is confirmed by the registration number. On certain products, the abbreviation, IC, before the registration number signifies that registration was performed based on a Declaration of Conformity indicating that Industry Canada technical specifications were met. It does not imply that Industry Canada approved the equipment.

AVIS: Le présent matériel est conforme aux spécifications techniques d'Industrie Canada applicables au matériel terminal. Cette conformité est confirmée par le numéro d'enregistrement. Le sigle IC, placé devant le numéro d'enregistrement, signifie que l'enregistrement s'est effectué conformément à une déclaration de conformité et indique que les spécifications techniques d'Industrie Canada ont été respectées. Il n'implique pas qu'Industrie Canada a approuvé le matériel.

NOTICE: The Ringer Equivalence Number (REN) for this terminal equipment is 0.4. The REN is an indication of the number of devices allowed to be connected to a telephone interface. The termination on an interface can consist of any combination of devices subject only to the requirement that the sum of the RENs of all the devices do not exceed 5.

AVIS: L'indice d'équivalence de la sonnerie (IES) du présent matériel est de 0.4. L'IES assigné à chaque dispositif terminal indique le nombre maximal de terminaux qui peuvent être raccordés à une interface téléphonique. La terminaison d'une interface peut consister en une combinaison quelconque de dispositifs, à la seule condition que la somme d'indices d'équivalence de la sonnerie de tous les dispositifs n'excède pas 5.

Before installing this equipment, users should ensure that it is permissible to be connected to the facilities of the local telecommunications company. The equipment must also be installed using an acceptable method of connection. The customer should be aware that compliance with the above conditions might not prevent degradation of service in some situations.

Repairs to certified equipment should be coordinated by a representative designated by the supplier. Any repairs or alterations made by the user to this equipment, or equipment malfunctions, may give the telecommunications company cause to request the user to disconnect the equipment.

Users should ensure for their own protection that the electrical ground connections of the power utility, telephone lines and internal metallic water pipe system, if present, are connected together. This precaution may be particularly important in rural areas.

Users should not attempt to make such connections themselves, but should contact the appropriate electric inspection authority, or electrician, as appropriate.

Australia

Fail-Safe Operation (AS/ACIF S038-2001).

This equipment will be inoperable when mains power fails.

This equipment will not continue to support emergency dialing after loss of mains power.

Safety Compliance Statements

The following customer information must be provided to customers with each product.

The statements provided below should be conspicuously located in the end user system documentation:

United States of America and Canada

This product is listed by Underwriters Laboratories, a Nationally Recognized Test Laboratory (NRTL). The Listing Mark is located on the bottom surface of the board. The Dialogic® Brooktrout® board has been tested and complies with UL 60950 and CAN/CSA-C22.2 No. 60950, Safety of Information Technology Equipment, Including Electrical Business Equipment.

Install this card only in UL Listed equipment that has instructions stating that the user may install and remove accessory cards.

This product must be mounted in the final assembly so that it is isolated from exposure to any hazardous voltages (voltages greater than 42.4V peak or 60Vdc) within the assembly. Adequate separation and restraint of cables and cords must be provided.

To maintain the safety certification of the system, ensure that the power drawn from the power supply does not exceed its capacity. Please refer to the power usage table on the hardware installation card applicable to your board for information on the voltages and currents required for proper operation.

Models of this card that contain DID interfaces are for use only in equipment that has a permanent connection to protective earth and is installed in a restricted access location.

Users should ensure for their own protection that the electrical ground connections of the power utility, telephone lines and internal metallic water pipe system, if present, are connected together. This precaution may be particularly important in rural areas.

Users should not attempt to make such connections themselves, but should contact the appropriate electric inspection authority, or electrician, as appropriate.

Disconnect any TNV circuit connectors (telephone line cords) from this card before removing the cover of the equipment.

Appendix B
Configuration from a Custom Installation Program

This appendix describes how one can create a custom installation program for manual configuration of the driver.

You can find installation structure and functions for the following operating systems:

- ***Linux Operating System on page 270***
- ***Solaris Operating System on page 274***
- ***Windows® Operating System on page 278***

Linux Operating System

You can perform configuration from a custom installation program by writing a program that runs ***dinstall*** with the `-c` option, or by writing a Bourne shell script that calls the provided installation functions described below.

Only write custom installation programs using one of these two mechanisms. Do not write them by directly including and/or modifying any of the supplied source or by writing new code to perform the same underlying system actions. Compatibility with installation functions for later versions cannot be guaranteed, but the up-to-date usage of the functions will always be documented.

The Bourne shell script containing the installation functions is ***dinstlib***, located in ***usr/sys/brooktrout/boston/driver/linux/install***. ***Dinstall*** is a particular example of a shell script that makes use of ***dinstlib***.

The functions generally use shell variables to pass values in and out. An exhaustive description of these variables is not supplied here. Use the Linux driver installation and the ***dinstall*** script as guides (see ***Chapter , Installing the Developer Kit on Red Hat Linux on page 9***).

Linux Installation

`mill_driver_install`

Required Input Variables

`Reserved`	
`Reserved`	
`Reserved`	
`max_pci_hw_modules`	Maximum number of PCI/cPCI hardware modules.
`physbufsize`	Physical buffer size.
`appbufsize`	Application buffer size.
`machine_id`	Machine ID, in hex (no leading 0x).
`do_reset`	Reset on driver startup.
`history_enable`	Whether to enable history.

`dhsize`	History size.
`num_hists_phy`	Number of physical histories.
`num_hists_apl`	Number of application histories.
`hist_phy_admin_only`	Whether to restrict physical histories to admin channels only.
`num_I2O_frames_per_module`	Number of I2O frames per module.
`fc_interval`	Driver flow control interval, in ms.
`mem_drv_alloc_min`	Min memory allocated per allocation.
`mem_drv_alloc_quanta`	Unit of allocation increment above min.
`num_reserved_modules`	Number of reserved modules.
`init_mod_vals`	A string consisting of the hex representation of pairs of module IDs and module numbers, each with the 0x prefix. Each pair is listed with a comma in between, and commas between each pair. If none, should be set to 0.
`mill_test1`	Driver test value 1
`mill_test2`	Driver test value 2
`mill_test3`	Driver test value 3
`driv_kvers_dir`	Location of all the ***boston.o*** binaries. This is normally the directory ***driver/linux/kernel/kvers***.

Output Variable

`err_msg` Contains displayable error message.

Note: If echo is used to display the contents, it might require the -e option to properly display some of the messages.

This variable is not modified except to set to 1 or error message.

Exit Code

0 = success
1 = error

Checking the Installation

`_mill_driver_install_check`

Exit Code

0 = not installed
non-0 = installed

Defining Default Symbols

`_mill_driver_defaults`

Defines the symbols that provide defaults for use with getting parameter values.

`DEFAULT_MOD_NUM`	Default starting reserved module number.
`DEFAULTAPPBUFSIZE`	Default application buffer size.
`Reserved`	Default ISA interrupt line. Reserved, no longer in use.
`DEFAULTMACHINEID`	Default machine ID.
`DEFAULTMAXHWMOD`	Default maximum number of PCI/cPCI modules.
`Reserved`	Default number of ISA modules. Reserved, no longer in use.
`DEFAULTPHYBUFSIZE`	Default physical buffer size.
`DHSIZEDEF`	Default history size.
`HIST_PHY_ADMIN_ONLY_DEF`	Default for physical histories admin only.
`MAX_RESERVED_MODULE_NUMS`	Maximum number of module reservations.
`MAXAPPBUFSIZE`	Maximum application buffer size.
`MAXMACHINEID`	Maximum allowable machine ID.
`Reserved`	Maximum number of ISA modules. Reserved, no longer in use.
`MAXPHYBUFSIZE`	Maximum physical buffer size.
`MINAPPBUFSIZE`	Minimum application buffer size.
`MINMACHINEID`	Minimum allowable machine ID.
`MINPHYBUFSIZE`	Minimum physical buffer size.
`NUM_HISTS_APL_DEF`	Default number of application histories.
`NUM_HISTS_PHY_DEF`	Default number of physical histories.

Removing the Driver

`_mill_driver_remove`

Input Argument

`partial` (Optional) The function only performs a partial removal. Also used for removal prior to reinstallation.

Output Variable

`err_msg` Contains displayable error message.

Note: If echo is used to display the contents, it might require the `-e` option to properly display some of the messages.

This variable is not modified except to set to 1 or error message.

Exit Code

0 = success
1 = error

Solaris Operating System

You can perform configuration from a custom installation program by writing a program that runs ***dinstall*** with the `-c` option, or by writing a Bourne shell script that calls the provided installation functions described below.

Only write custom installation programs using one of these two mechanisms. Do not write them by directly including and/or modifying any of the supplied source or by writing new code to perform the same underlying system actions. Compatibility with installation functions for later versions cannot be guaranteed, but the up-to-date usage of the functions will always be documented.

The Bourne shell script containing the installation functions is ***dinstlib***, located in ***boston/driver/solaris/install***. ***Dinstall*** is a particular example of a shell script that makes use of ***dinstlib***.

The functions generally use shell variables to pass values in and out. An exhaustive description of these variables is not supplied here. Use the Solaris driver installation and the ***dinstall*** script as guides (see ***Installing the Software on page 29*** and ***Configuring the Driver Manually on page 33***).

Solaris Installation

`mill_driver_install`

Required Input Variables

`Reserved`	0
`Reserved`	0
`Reserved`	0
`max_pci_hw_modules`	Maximum number of PCI/cPCI hardware modules
`physbufsize`	Physical buffer size.
`appbufsize`	Application buffer size.
`machine_id`	Machine ID, in hex (no leading 0x).
`do_reset`	Reset on driver startup.
`history_enable`	Whether to enable history.
`dhsize`	History size.

`num_hists_phy`	Number of physical histories.
`num_hists_apl`	Number of application histories.
`hist_phy_admin_only`	Whether to restrict physical histories to admin channels only.
`num_I2O_frames_per_module`	Number of I2O frames per module.
`fc_interval`	Driver flow control interval, in ms.
`mem_drv_alloc_min`	Min memory allocated per allocation.
`mem_drv_alloc_quanta`	Unit of allocation increment above min.
`num_reserved_modules`	Number of reserved modules.
`init_mod_vals`	A string consisting of the hex representation of pairs of module IDs and module numbers, each with the 0x prefix. Each pair is listed inside braces [] with a comma in between, and commas between each pair. If none, set to 0.
`mill_test1`	Driver test value 1
`mill_test2`	Driver test value 2
`mill_test3`	Driver test value 3
`driv_bin_file`	Location of the 32-bit ***boston*** binary.
`driv_bin64_file`	Location of the 64-bit ***boston*** binary, where applicable.
`driv_unload_file`	Location of the ***bostunld*** binary.
`driv_arch_file`	Location of the ***bost_arch*** file, normally ***driver/solaris/kernel/bost_arch***.

Output Variables

`err_msg`	Contains displayable error message.

This variable is not modified except to set to 1 or error message.

Exit Code

0 = success
1 = error

Checking the Installation

`_mill_driver_install_check`

Exit Code

0 = not installed
non-0 = installed

Defining Default Symbols

`_mill_driver_defaults`

Defines the symbols that provide defaults for use with variable values.

DEFAULT_MOD_NUM	Default starting reserved module number.
DEFAULTAPPBUFSIZE	Default application buffer size.
Reserved	Default ISA interrupt line. Reserved, no longer in use.
DEFAULTMACHINEID	Default machine ID.
DEFAULTMAXHWMOD	Default maximum number of PCI/cPCI modules.
Reserved	Default number of ISA modules. Reserved, no longer in use.
DEFAULTPHYBUFSIZE	Default physical buffer size.
DHSIZEDEF	Default history size.
HIST_PHY_ADMIN_ONLY_DEF	Default for physical histories admin only.
MAX_RESERVED_MODULE_NUMS	Maximum number of module reservations.
MAXAPPBUFSIZE	Maximum application buffer size.
MAXMACHINEID	Maximum allowable machine ID.
Reserved	Maximum number of ISA modules. Reserved, no longer in use.
MAXPHYBUFSIZE	Maximum physical buffer size.
MINAPPBUFSIZE	Minimum application buffer size.
MINMACHINEID	Minimum allowable machine ID.
MINPHYBUFSIZE	Minimum physical buffer size.
NUM_HISTS_APL_DEF	Default number of application histories.
NUM_HISTS_PHY_DEF	Default number of physical histories.

Removing the Driver

`_mill_driver_remove`

Input Argument

`partial` (Optional) The function only performs a partial removal. Also used for removal prior to reinstallation.

Output Variable

`err_msg` Contains displayable error message.

This variable is not modified except to set to 1 or error message.

Exit Code

0 = success
1 = error

Windows® Operating System

You can perform configuration from a custom installation program by writing a program that runs ***install*** with the `-c` option, or by writing a program that calls the Dialogic-provided installation functions, described below, that can be compiled and linked into the program.

Only write custom installation programs using one of these two mechanisms. Do not write them by directly including and/or modifying any of the supplied source or by writing new code to perform the same underlying system actions. Compatibility with installation functions for later versions cannot be guaranteed, but the up-to-date usage of the functions will always be documented.

All files referred to are in the ***brooktrout\boston\driver\winnt\install*** directory. To use these functions, the custom program must include ***instlib.obj*** and ***pnplib.obj***. The object files must be linked with the custom program. ***Install.c*** is a particular example of a program that is compiled and linked in this way.

Note: To build a custom program with these object files requires the Windows® DDK (2000 or higher).

An exhaustive description of the function parameters is not supplied here. Use the Windows® driver configuration and the ***install.c*** program as guides (see ***Chapter , Installing the Device Driver on Windows® on page 57***).

Installing the Windows® Driver

Use the following function to install the Windows® PnP driver:

```
int _mill_pnp_driver_install(
   struct driver_parameters *drv_parm,
   int    do_adjustments,
   int    already_option,
   int    meg_in_system,
   int    *mem_adjust_stat,
   char   *inf_file,
   int    load_uncond,
   int    *reboot_required,
   char   **failed_win32_call,
   DWORD  *win32_err_val,
   char   **other_msg
);
```

Input Variables

`already_option`	Action to take if already adjusted.
`appbufsize`	Application buffer size.
`dest_dir_name`	Pointer to `char *` variable to contain the directory name in which the boston driver files were installed.
`dhsize`	History size.
`do_adjustments`	Whether to do PCI memory adjustments.
`do_reset`	Reset on driver startup.
`driver_fname`	Name of driver binary ***boston.sys*** file to install.
`drv_parm`	Pointer to a structure containing driver parameters.
`failed_win32_call`	Pointer to `char` * variable to contain the name of a Windows® function that failed.
`fc_interval`	Driver flow control interval, in ms.
`hist_phy_admin_only`	Whether to restrict physical histories to administrative channels only.
`history_enable`	Whether to enable history.
`inf_file`	Name of driver ***.inf*** file.
`Reserved`	ISA interrupt line is no longer used.
`load_uncond`	Indicates that driver should load regardless of devices being present in the system.
`machine_id`	Machine ID.
`max_pci_hw_modules`	Maximum number of PCI/cPCI hardware modules.

`meg_in_system`	Amount of memory in system.
`mem_adjust_stat`	Pointer to `int` variable to contain result of memory adjustment attempt 0 = Memory adjustment not performed 1 = Memory adjustment performed 2 = Previous adjustment detected, action dependent on `already_option`
`mem_drv_alloc_min`	Min memory allocated per allocation.
`mem_drv_alloc_quanta`	Unit of allocation increment above min.
`Reserved`	Pointer to array of ISA addresses is no longer used.
`Reserved`	Size of `mill_isa_addrs` array is no longer used.
`mill_test1`	Driver test value 1.
`mill_test2`	Driver test value 2.
`mill_test3`	Driver test value 3.
`num_hists_apl`	Number of application histories.
`num_hists_phy`	Number of physical histories.
`num_I2O_frames_per_module`	Number of I2O frames per module.
`Reserved`	Number of ISA modules is no longer used.
`num_reserved_modules`	Number of reserved modules.
`other_msg`	Pointer to `char *` variable to contain a possible error message.
`phybufsize`	Physical buffer size.
`pkt_version`	Reserved, must be 0.
`reboot_required`	Pointer to `int` variable to contain indication of whether a reboot is required.
`reserved_modules`	Pointer to array of reserved module information.
`reserved_modules_size`	Size of reserved_modules array.
`win32_err_val`	Win32 error code returned by failing Windows® function.

Return Value

0 = success
1 = error

Driver Parameters Structure

Use the `driver_parameters` structure to install the PnP driver (see ***Installing the Windows® Driver on page 279***).

For input variable definitions, see ***Input Variables on page 279***.

```
struct driver_parameters
{
   int      reserved;
   int      reserved;
   unsigned reserved;
   unsigned reserved;
   unsigned max_pci_hw_modules;
   unsigned phybugsize;
   unsigned appbufsize;
   unsigned machine_id;
   unsigned do_reset;
   unsigned history_enable;
   unsigned dhsize;
   unsigned num_hists_phy;
   unsigned num_hists_apl;
   unsigned hist_phy_admin_only;
   unsigned num_I20_frames_per_module;
   unsigned fc_interval;
   unsigned mem_drv_alloc_min;
   unsigned mem_drv_alloc_quanta;
   unsigned pkt_version;
   unsigned num_reserved_modules;
   struct reserved_module_num *reserved_modules;
   unsigned reserved_modules_size;
   unsigned mill_test1;
   unsigned mill_test2;
   unsigned mill_test3;
};
```

Checking the Installation

Verifies that the driver is installed.

```
int _mill_pnp_driver_install_check(void);
```

Return Value

0 = not installed
1 = installed

Defining Default Symbols

The ***instlib.h*** file defines the following symbols that provide defaults for use with getting parameter values.

You can find the ***instlib.h*** file at the following location:

```
C:\Brooktrout\Boston\driver\winnt\install\
```

DEFAULT_MOD_NUM	Default starting reserved module number.
DEFAULTAPPBUFSIZE	Default application buffer size.
DEFAULTINTRLINE	Default ISA interrupt line. Reserved, no longer in use.
DEFAULTMACHINEID	Default machine ID.
DEFAULTMAXHWMOD	Default maximum number of PCI/cPCI modules.
DEFAULTNMILL	Default number of ISA modules. Reserved, no longer in use.
DEFAULTPHYBUFSIZE	Default physical buffer size.
DFLT_FC_INTERVAL	Default number of FC intervals.
DFLT_I2O_FRAMES_PER_MODULE	Default number of I2O frames per module.
DHSIZEDEF	Default history size.
DHSIZEMIN	Minimum history size.
HIST_PHY_ADMIN_ONLY_DEF	Default for physical histories admin only.
I2O_FRAMES_PER_MODULE	Number of I2O frames per module.
MAX_RESERVED_MODULE_NUMS	Maximum number of module reservations.
MAXAPPBUFSIZE	Maximum application buffer size.
MAXMACHINEID	Maximum allowable machine ID.
MAXNMILL	Maximum number of ISA modules. Reserved, no longer in use.
MAXPHYBUFSIZE	Maximum physical buffer size.

MEM_DRV_ALLOC_MIN	Minimum allocation of memory.
MEM_DRV_ALLOC_QUANTA	Number of allocation of memory.
MIN_FC_INTERVAL	Minimum number of FC intervals.
MINAPPBUFSIZE	Minimum application buffer size.
MINMACHINEID	Minimum allowable machine ID.
MINPHYBUFSIZE	Minimum physical buffer size.
NUM_HISTS_APL_DEF	Default number of application histories.
NUM_HISTS_PHY_DEF	Default number of physical histories.

This file also defines `SYSDIR`, a string that represents the type of platform being compiled on, and a component of the subdirectory path needed to find the compiled driver binary.

It also provides the `struct reserved_module_num` definition, used by `_mill_pnp_driver_install`.

Removing the Driver

Uninstalls the driver service and removes files. For input variable definitions, see ***Input Variables on page 279***.

```
int _mill_pnp_driver_remove(
   int    *mem_adjust_stat,
   int    *reboot_required,
   char   **failed_win32_call,
   DWORD  *win32_err_val,
   char   **other_msg
);
```

Return Value

0 = success
1 = error

Appendix C
Reinitializing the Device Driver

This appendix describes the driver reinitialization utility. This utility is used to return the device driver to a state as close as possible to that of a fresh driver start without stopping the driver.

Introducing the Driver Reinitialization Utility

The driver can be reinitialized close to its starting state in terms of channels assigned to modules and module number assignments using the ***driver_reinit*** program located in the ***boston/driver/<platform>/user*** directory.

Use of this program may avoid the need to stop and restart the driver. This is especially useful for a Windows® PnP system as the driver cannot be easily started and stopped.

Using the driver_reinit Utility

When you execute the ***driver_reinit*** program, this utility will return the driver to a state as close as possible to that of a fresh driver start without stopping the driver.

The utility will:

- Clear all ordinal channel assignments
- Remove all stored information about the modules
- Reset all the hardware modules

The utility fails if any application sessions currently exist.

Glossary

API Application Programming Interface

ASR Automatic Speech Recognition

Channel A logical channel of operations provided by a Boston module. See ***logical channel number***, ***ordinal channel number***, ***work channel***.

EC Echo cancellation

Facility A software entity responsible for a set of related functions that provide services to the host, e.g., fax facility and voice facility.

ISDN Integrated Services Digital Network

IVR Interactive Voice Response

lapdid The term ***lapdid*** has its origins with the LAP-D protocol used for call control, but has an extended meaning for Dialogic products. For call control with the ISDN protocol (Q.931), ***lapdid*** refers to a trunk-specific HDLC controller (trunk 1 = lapdid 0, trunk 2 = lapdid 2, trunk 3 = lapdid 4 and trunk 4 = lapdid 6). For call control with the LEC protocols (RBS, etc.), ***lapdid*** refers to a specific 0-based trunk number (trunk 0 = lapdid 0, trunk 1 = lapdid 1,..., trunk 6 = lapdid 6 and trunk 7 = lapdid 7).

Line A T1/E1 slot or a single analog slot. Lines are numbered starting at 1.

Logical channel number A number used with the hardware module number to reference a channel in a system. Channel 0 is reserved and channel 1 is the administrative channel. Other channels are numbered from 2 to n+1 (where n is the number of work channels defined for the module).

Millennium Address An address of a communicating BOSTON or Millennium entity, a facility and channel on a BOSTON module, or the BOSTON device driver. Each address consists of 4 components, the facility, the channel, the module, and the machine. It is possible to make use of most features of the firmware and the APIs without requiring direct use of these addresses.

Module A communicating Boston entity that usually represents a CPU on a Dialogic® Brooktrout® board, a host application, a software telephony module (SR140) or the Boston driver. See ***Determining the Board Module Number on page 2*** for more information about modules and module number assignment.

NVRAM Non-Volatile RAM. Random Access Memory that is not erased when the board is powered off or when the firmware is downloaded again.

Ordinal channel number A number in the range 0 to n-1 where n is the total number of work channels on all the boards in the system.

Packet In the BOSTON architecture, a sequence of bytes containing a BOSTON destination address, source address, and one or more commands.

PCI Peripheral Component Interconnect

PDF Portable Document Format

Plug-and-Play Hardware or software that, after installation, can be used immediately without configuration. Also PnP.

Port A TDM bus (for example: MVIP, H110), DS-1 (T1/E1), BRI or analog interface.

Stream A logical data entity that corresponds to a physical data line on a TDM bus

T1/E1 span The set of slots that comprises one T1 (24) or one E1 (30) line. The spans are numbered starting at 1.

TCP Transmission Control Protocol

Time slot A logical entity that corresponds to one telephone call.

Unit number 1. In telephony configuration files and functions, a unit is a hardware port on a Dialogic® Brooktrout® board and is numbered starting at 0 (to denote a TDM bus); from 1 to n (for a specific T1/E1 interface).

2. In some Bfv functions (for example, ***BfvLineAttach***), an ordinal channel number. Its range is 0...n-1, where n is the number of channels in a system.

VAD Voice Activity Detection

Work channel One of the channels on a module that is available for non-administrative purposes as a result of downloading firmware. A module configured for 48 channels when firmware download is performed has 48 work channels and one administrative channel. Logical work channel numbers start at 2 on each module.

Index

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W