Dialogic® Diva® Management API

Developer’s Reference Guide

Part of the Dialogic® Diva® Software Development Kit

www.dialogic.com

Seventh Edition (October 2007) 206-409-07

Copyright and Legal Disclaimer

All contents of this document are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation or its subsidiaries. Reasonable effort is made to ensure the accuracy of the information contained in the document. However, due to ongoing product improvements and revisions, Dialogic Corporation and its subsidiaries do not warrant the accuracy of this information and cannot accept responsibility for errors or omissions that may be contained in this document.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH DIALOGIC® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

It is possible that the use or implementation of any one of the concepts, applications, or ideas described in this document, in marketing collateral produced by or on web pages maintained by Dialogic Corporation or its subsidiaries may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic Corporation or its subsidiaries do not provide any intellectual property licenses with the sale of Dialogic products other than a license to use such product in accordance with intellectual property owned or validly licensed by Dialogic Corporation or its subsidiaries. More detailed information about such intellectual property is available from Dialogic Corporation's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. The software referred to in this document is provided under a Software License Agreement. Refer to the Software License Agreement for complete details governing the use of the software.

Dialogic Corporation encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.

Dialogic and Diva are registered trademarks of Dialogic Corporation or its subsidiaries. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic's trademarks requires proper acknowledgement.

Microsoft, Visual C++ and Windows are registered trademarks of Microsoft Corporation in the United States and/or other countries. The names of actual companies and products mentioned herein are the trademarks of their respective owners.

To contact Dialogic Customer Support, visit our Web site at www.dialogic.com/support.

This software is based in part on the work of the Independent JPG Group.

Dialogic Corporation License Agreement for Use of Software

This is an Agreement between you, the Company, and your Affiliates (referred to in some instances as "You" and in other instances as "Company") and all your Authorized Users and Dialogic Corporation ("Dialogic").

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE INSTALLING OR DOWNLOADING THE SOFTWARE. IF YOU AGREE WITH THESE TERMS YOU MAY PROCEED WITH THE DOWNLOAD OR INSTALLATION OF THE SOFTWARE. IF YOU DO NOT AGREE WITH THESE TERMS, PLEASE RETURN THE PACKAGE IN "AS NEW" CONDITION (INCLUDING DOCUMENTATION AND BINDERS OR OTHER CONTAINERS) AND YOUR MONEY WILL BE REFUNDED. DOWNLOADING OR INSTALLING THE SOFTWARE CONSTITUTES YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS. YOU ASSUME RESPONSIBILITY FOR THE SELECTION OF THE PROGRAM TO ACHIEVE YOUR INTENDED RESULTS, AND FOR THE INSTALLATION, USE, AND RESULTS OBTAINED FROM THE PACKAGE.

Copyright

The enclosed Software ("Program") and documents are owned by Dialogic Corporation ("Dialogic") and its suppliers and are protected by copyright laws and international treaty provisions. Therefore, You and your Authorized Users must treat the Program and documentation like any other copyrighted material except as expressly permitted in this License Agreement.

License

Under the terms and conditions of this License Agreement:

- You may install and use one copy of the Program on a single-user computer, file server, or on a workstation of a local area network, and only in conjunction with a legally acquired Dialogic hardware or software product;
- The primary Authorized User on the computer on which the "Program" is installed may make a second copy for his/her exclusive use on either a home or portable computer;
- You may copy the Program into any machine readable or printed form for backup or modification purposes in support of your use of one copy of the Program;
- You may make one copy of Dialogic's documentation provided that all copyright notices contained within the documentation are retained;
- You may modify the Program and/or merge it into another Program for your use in one computer; (any portion of this Program will continue to be subject to the terms and conditions of this Agreement);
- You may transfer the Program, documentation and the license to another eligible party within your Company if the other party agrees to accept the terms and conditions of this Agreement. If You transfer the Program and documentation, You must at the same time either transfer all copies whether in printed or machine readable form to the same party or destroy any copies not transferred; this includes all modifications and portions of the Program contained in or merged into other Programs;
- You must reproduce and include the copyright notice on any copy, modification or portion of the Program merged into another Program;
- You may not rent or lease the Program. You may not reverse engineer, decompile or disassemble the Program. You may not use, copy, modify or transfer the Program and documentation, or any copy, modification or merged portion, in whole or in part, except as expressly provided for in this License Agreement;
- If You transfer possession of any copy, modification or merged portion of the Program or documentation to another party in any way other than as expressly permitted in this License Agreement, this license is automatically terminated.

Upgrades

If the Program is provided as an upgrade and the upgrade is an upgrade from another software product licensed to You and Your Authorized Users by Dialogic, the upgrade is governed by the License Agreement earlier provided with that software product package and the present License Agreement does not grant you additional license(s).

Term

The license is effective until terminated. You may terminate it at any time by destroying the Program and documentation together with all copies, modifications and merged portions in any form. It will also terminate upon conditions set forth elsewhere in this Agreement or if you fail to comply with any terms or conditions of this Agreement. You agree upon such termination to destroy the Program and documentation together with all copies, modifications and merged portions in any form.

Limited Warranty

The only warranty Dialogic makes is that the medium on which the Program is recorded will be replaced without charge if Dialogic, in good faith, determines that it was defective in materials or workmanship and if returned to your supplier with a copy of your receipt within ninety (90) days from the date you received it. Dialogic offers no warranty for your reproduction of the Program. This Limited Warranty is void if failure of the Program has resulted from accident, misuse, abuse, or misapplication.

Customer Remedies

Dialogic's entire liability and You and Your Authorized Users exclusive remedy shall be, at Dialogic's option, either (a) return of the price paid or (b) repair or replacement of the Program that does not meet the above Limited Warranty. Any replacement Program will be warranted for the remainder of the original Warranty period.

No Other Warranties

Dialogic disclaims all other warranties, either expressed or implied, including but not limited to implied warranties or merchantability and fitness for a particular purpose and the warranty against latent defects, with respect to the Program and the accompanying documentation. This limited warranty gives You specific legal rights. You may have others, which may vary from jurisdiction to jurisdiction.

No Liability for Consequential Damage

In no event shall Dialogic or its suppliers be liable for any damages whatsoever (including without limitation, damages for loss of business profits, business interruption, loss of information, or other pecuniary loss and indirect, consequential, incidental, economic, or punitive damages) arising out of the use of or inability to use this Program, even if Dialogic has been advised of the possibility of such damages. As some jurisdictions do not allow the exclusion or limitation for certain damages, some of the above limitations may not apply to You or your Authorized Users.

Limit of Liability

Dialogic's entire aggregate liability under any provision of this agreement shall be limited to the amount actually paid by You for the affected Program.

Right to Audit

If this Program is licensed for use in a Company, your Company agrees to keep all usual and proper records and books of accounts and all usual proper entries relating to each reproduction and Authorized User of the Program during the term of this Agreement and for a period of three (3) years thereafter. During this period, Dialogic may cause an audit to be made of the applicable records in order to verify Your compliance with this Agreement and prompt adjustment shall be made to compensate for any errors or omissions disclosed by such audit. Any such audit shall be conducted by an independent certified public accountant selected by Dialogic and shall be conducted during the regular business hours at Your offices and in such a manner as not to interfere with Your normal business activities. Any such audit shall be paid for by Dialogic unless material discrepancies are disclosed. For such purposes, "material discrepancies" shall mean three percent (3%) or more of the Authorized Users within the Company. If material discrepancies are disclosed, Your Company agrees to pay Dialogic for the costs associated with the audit as well as the license fees for the additional Authorized Users. In no event shall audits be made more frequently than semi-annually unless the immediately preceding audit disclosed a material discrepancy.

Supplementary Software

Any Supplementary Software provided with the Dialogic Program referred to in this License Agreement is provided "as is" with no warranty of any kind.

U.S. Government Restricted Rights

The Program and documentation are provided with RESTRICTED RIGHTS. Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph c) 1) ii) of The Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraph c) 1) and 2) of the Commercial Computer Software-Restricted Rights at 48 CFR52.227-19, as applicable.

Governing Law

This Agreement shall be construed and controlled by the laws in force in the Province of Quebec, Canada.

Contractor/ manufacturer is:

Dialogic CORPORATION.

9800 Cavendish Blvd., Montreal, Quebec, Canada H4M 2V9

This Agreement has been drafted in English at the express wish of the parties. Ce contrat a été rédigé en anglais à la demande expresse des parties.

Contents

CHAPTER 1

About this Online Guide

How to use this online guide

- To view a section, click the corresponding bookmark located on the left.
- To view a topic that contains further information, click the corresponding blue underlined phrase.

Structure of this guide

This guide presents implementation details and functional descriptions of all commands in the Dialogic® Diva® API Library interface. Examples are provided where applicable. Constants, data structures, and return codes are described in the appendices.

This guide is structured as follows:

Section	Contents
Dialogic® Diva® SDK Overview	Introduction to the Diva software development kit and its five application programming interfaces, the Dialogic® Diva® Component API, Dialogic® Diva® API, extended CAPI 2.0, Dialogic® Diva® Management API, and Dialogic® Diva® API for .NET
Dialogic® Diva® Management API Overview	Introduction to the Diva Management API, outline of the information that can be retrieved via the Diva Management API, prerequisites for using the Diva Management API
Functions	Description of the C functions used to access the Diva Management API
Nodes and Values	Description of the nodes and values that can be read out by the Diva Management API

CHAPTER 2

Dialogic® Diva® SDK Overview

The Dialogic Diva SDK can be used in combination with Dialogic® Diva® Media Boards and Dialogic® Host Media Processing (HMP) software. On these communication platforms, the Diva SDK provides the following application programming interfaces (APIs): the Dialogic® Diva® API, the Dialogic® Diva® Components API, and the Dialogic® Diva® API for .NET. For the Diva Media Boards, two additional APIs are available: the Dialogic® Diva® Management API and the Extended CAPI 2.0.

It is planned that new versions of the Diva SDK will be released periodically, and it is intended that such new versions will be backwards compatible so as to allow applications developed on the basis of earlier versions of the Diva SDK to be used with the new versions.

The Diva SDK includes the following components:

- Libraries providing functions to access the Dialogic® Diva® communication platforms
- DLLs containing the interfaces and component services
- Programming samples in source code
- Documentation explaining the functions of the Diva SDK

The components can be found as follows:

Component	Path
Libraries of the Diva SDK	\SDK\BASIC\LIB\
DLLs of the Diva SDK and compiled samples applications	\SDK\BASIC\BIN
Samples for the Diva SDK	\SDK\BASIC\SAMPLES\
Libraries of the Diva Management API	\SDK\MANAGEMENT\LIB
Samples for the Management API	\SDK\MANAGEMENT\SAMPLES
Documentation	\SDK\DOC\

The Diva SDK is available on the Dialogic® Diva® Software Suite CD-ROM. You can also download it from the Dialogic web site under http://www.dialogic.com/products/tdm_boards/development_tools/default.htm. If you download the software from the Dialogic web site, extract the files to your hard disk and do not change the directory structure of the extracted files.

The Diva SDK is freely distributed with Dialogic® communication platforms. You do not have to purchase licences for developing applications based on the software development kit.

Dialogic® Diva® SDK application programming interfaces

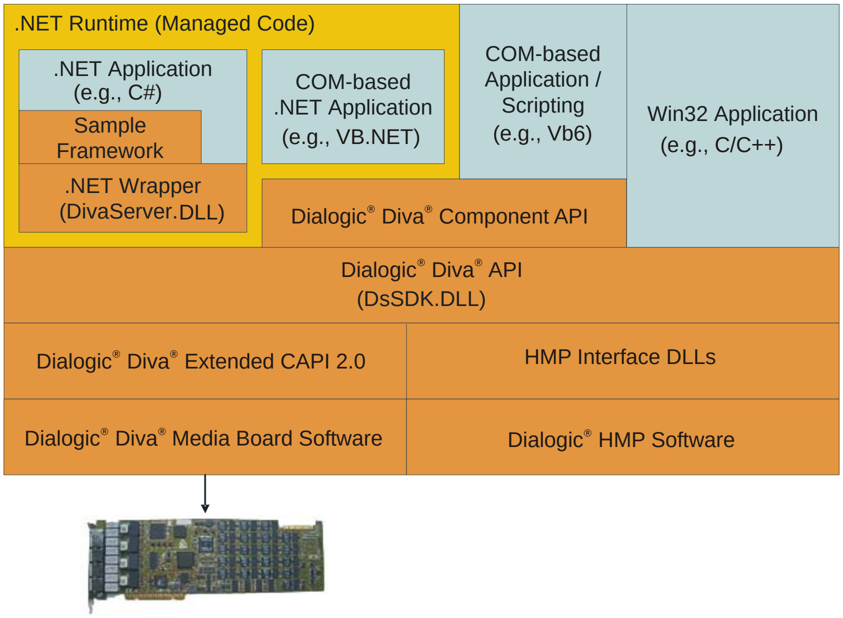
The five application programming interfaces (APIs) of the Dialogic® Diva® SDK represent different layers for the management and development of applications for Dialogic® Diva® communication platforms.

- Dialogic® Diva® API: It provides a high-level interface into the communication platforms that allows developers to implement communication applications. It also provides an additional library for data conversion like TIFF to SFF for fax applications.
- Dialogic® Diva® Component API: It provides a set of ActiveX components that allows developers to create new applications or to add telephony and communication features to existing applications. The Component API can be used from scripts and VB.NET and eliminates the need to write directly to a C / C++ API.
- Dialogic® Diva® API for .NET: It is based on the Dialogic® Diva® API and provides access to the Diva API from .NET-based applications.
- Extended CAPI 2.0 (only for Dialogic® Diva® Media Boards): It provides Dialogic-specific CAPI extensions that are fully CAPI 2.0 compliant.

- Dialogic® Diva® Management API: It provides direct hardware access for monitoring, security, and statistics. This API should only be used by applications using CAPI 2.0.

Dialogic® communication platforms provide call control, media streaming, and management functionality that are available on the Diva API, the Diva Component API, and the Diva API for .NET.

The following figure shows the architecture of the programming interfaces and the applications that may access them:


.NET Runtime (Managed Code)
.NET Application (e.g., C#)
Sample Framework
.NET Wrapper (DivaServer.DLL)
COM-based .NET Application (e.g., VB.NET)
COM-based Application / Scripting (e.g., Vb6)
Win32 Application (e.g., C/C++)
Dialogic® Diva® Component API
Dialogic® Diva® API (DsSDK.DLL)
Dialogic® Diva® Extended CAPI 2.0
HMP Interface DLLs
Dialogic® Diva® Media Board Software
Dialogic® HMP Software

Dialogic® Diva® API

The Diva API is a high-level interface into the Diva communication platforms via a library of "C" function calls. This interface can allow developers to implement various communication applications faster and easier than in the traditional CAPI 2.0 application development.

The Diva API contains modules that can be used as basis for communication applications, such as fax and voice transfer or call control, and thus can facilitate the development of applications for these areas. The modules are intended to be updated so as to offer development bases for an ever increasing range of communication applications.

Even if the Diva API abstracts functions and provides a high level interface, access to low level functions is optionally available. Applications that require access to low level operations, e.g., control over signaling messages, can be performed on the Diva API. This allows existing applications to be extended using the same API, even if the requirements change. All CAPI 2.0 extensions are also available on the Diva API.

The Diva API also allows for access to the management interface of the Dialogic® Diva® Media Board for status and statistic information.

Dialogic® Diva® Component API

The Diva® SDK includes the Diva Component API. This interface provides a set of ActiveX components that allow for creating new applications or for adding telephony and communication features to existing applications. The components can be used in Windows®-based development environments that support ActiveX components. The Diva Component API provides functionalities including: extended call control, voice streaming and recording, change of media on existing calls, conferencing, call transfer, retrieve status information of the hardware, flexible tone detection, and answering machine detection.

Dialogic® Diva® API for .NET

The Diva API for .NET is based on the Dialogic® Diva® API. The functionality is exactly the same and where possible a one to one mapping is done. The Diva API for .NET documentation describes the architecture, the mapping of data types, and the modifications that have been done due to the .NET and especially the C# requirements.

Applications written for .NET may be based on two APIs provided by the Diva SDK: the Dialogic® Diva® Component API and the Dialogic® Diva® API for .NET. Both APIs use the feature rich Diva API to access the underlying communication platform. The Diva Component API allows for synchronous processing and is therefore often used by script-oriented applications. The decision as to which Dialogic® Diva® API can be used depends on the application requirements.

Extended CAPI 2.0

The Extended CAPI 2.0 is only available for Dialogic® Diva® Media Boards and provides Dialogic-specific extensions for CAPI 2.0. The extensions are fully CAPI 2.0 compatible, and thus can be used with CAPI 2.0 applications. The following Dialogic-specific CAPI extensions are available:

- Echo canceller support for voice applications: This extension allows the voice application to place an echo canceller unit in the front end of a connection to suppress acoustical echo and signal return. The Dialogic extension and the new CAPI standard for echo canceller are supported.
- Extension for fax paper formats and resolutions: This extension enables fax transmission and reception with an extended range of paper formats and resolutions.
- Tone detection and generation extension for DTMF facility: This extension enables fax and voice applications to detect in-band signals such as busy tone, to report events like modem CNG or fax flag detection, to detect human speech, to report the unidentified tones, and to report that no signal is present on the line.
- Extensions for modem configuration: This extension enables to specify certain modulation and protocol-related parameters. Modulations can be removed from the auto moding list or specific modulations can be selected. The results of the modulation and the protocol negotiation are signaled to the application.
- Generic tone generator and detector support for voice applications: This extension provides built-in generic tone detector and generator facilities.The generic tone services include sine generators with programmable frequency and amplitude modulation, function generators with programmable signal shape, frequency, and amplitude modulation, noise generators with programmable crest factor and amplitude modulation, single tone detection, and dual tone detection.

Descriptions of the Dialogic-specific CAPI 2.0 extensions are available under `SDK/DOC`. The complete CAPI 2.0 specification can be downloaded from the web site www.capi.org.

Note: If you are developing CAPI 2.0 applications based on Dialogic® Diva® for Windows® software, the CAPI 2.0 is part of the Diva API in the Dialogic® Diva® Configuration Manager.

Dialogic® Diva® Management API

The Diva Management API is only available for Dialogic® Diva® Media Boards and only applications based on the CAPI 2.0 may use this interface. Applications using the Dialogic® Diva® API, the Dialogic® Diva® Component API, or Dialogic® Diva® API for .NET should not use this API.

The Diva Management API offers a range of functions to retrieve status and statistics information:

- Retrieve status of lowest level ISDN access
- Retrieve active calls
- Generating statistic information for number of calls, etc.
- Notification of status changes

Applications using the Dialogic® Diva® API do not need to access the Diva Management API, because the Diva API already has built-in support for the Diva Management API.

The Diva Management API only registers information and executes functions within your system. You cannot use it for data transfer. For data transfer, you should use the Diva API or CAPI interface.

The status information provided by the Diva Management API is structured as a sort of "virtual" file space. It contains nodes (similar to directories) and values (similar to files), where each node and value is defined by its path and name. The information can be read out using Diva Management API functions. The DLL providing these functions is part of the Dialogic® Diva® for Windows® software.

The functions provided by the Diva Management API are extended periodically; however, the Diva Management API will remain backwards compatible such that applications based on an earlier Diva Management API can continue to be used with new versions.

Dialogic communication platform-related information

The Dialogic® Diva® SDK uses the Dialogic® Diva® System Release software or the Dialogic® HMP System Release software to communicate to the TDM or IP-based communication resources. The Diva-based software is automatically started at system start and configured via the Dialogic® Diva® Configuration Manager.

The Dialogic HMP software is started either automatically when the system starts or manually depending on the configuration in the Dialogic® HMP Configuration Manager (DCM). With the initialization of the Dialogic® Diva® API by the application, the Dialogic HMP is initialized and configured. The configuration parameters are read from the file "dssdk.xml". Refer to the "Dialogic® HMP Software and Dialogic® Diva® SDK Installation and Configuration Guide" for details on those configuration parameters.

The Dialogic HMP features are based on licenses, and there are various options that can be combined. Based on the available licenses, Dialogic® Diva® API interface functions may return *DivaErrorNotSupported* if a requested function is not licensed or no more licenses are available. The following section provides detailed information.

The Diva API supports voice, conferencing, and fax on the Dialogic HMP software. In addition, mixed conferences between Dialogic® Diva® Media Boards and the Dialogic HMP software are supported. Tromboning, called Line Interconnect on the Diva API, is also supported between Diva Media Boards and the Dialogic HMP software.

The following license options are validated during startup of the Diva API:

- IP call control / RTP G.711
- voice
- speech integration
- conferencing
- fax

The Diva SDK allocates resources for the duration of a call. If a call is initiated as a voice call, a voice resource is allocated and assigned to this call. This resource remains allocated even if the application switches to fax mode later.

IP call control / RTP G.711 resources

The amount of IP call control / RTP G.711 resources specifies the maximum number of channels. When started, the Dialogic® Diva® API creates the virtual line devices based on the configuration information that is specified in the configuration file. By default, only one line device is created for SIP-based communication using the available channels.

If line devices are configured and the amount of configured channels exceeds the licensed channels, the amount of channels for a line device or the amount of line devices may be limited. The application detects it in the information returned by *DivaGetNumLineDevices* and *DivaGetLineDeviceInfo*.

Voice and speech integration resources

The Dialogic® HMP software provides voice resources for supporting features such as streaming audio, detecting DTMF tones, and generating DTMF tones. The capabilities of a voice resource depend on the license. If only "voice" is licensed, this resource can either play or record, but not both at the same time. A "speech integration" resource can play and record in parallel, and it supports an echo canceller. Based on the available voice resources, the Dialogic® Diva® API supports three different modes and the interface functions *DivaRecordVoiceFile* and *DivaSendVoiceFile* may behave differently. During system start, the Dialogic® Diva® SDK enumerates the Dialogic HMP software resources and selects one of the following three modes:

- Two voice resources per channel
- One speech integration voice resource per channel
- One voice resource per channel

Two voice resources per channel

This mode allows to play and record in parallel and it is entered if two voice resources are available for each licensed IP-channel. The behavior of the functions *DivaRecordVoiceFile* and *DivaSendVoiceFile* is the same as on Dialogic® Diva® Media Boards.

When a call is established, the received audio is signaled to the application via the event *DivaEventDataAvailable* and can be retrieved by the application via *DivaReceiveAudio*.

One speech integration voice resource per channel

This mode is entered if one speech integration license per IP-channel is available and it allows to play and to record in parallel. The functions *DivaRecordVoiceFile* and *DivaSendVoiceFile* behave like on Diva Media Boards. For received audio, the echo canceller can be enabled.

When a call is established, the received audio is signaled to the application via the event *DivaEventDataAvailable* and can be retrieved by the application via *DivaReceiveAudio*.

This mode allows for retrieving audio in small buffer sizes and is the base for bridging between TDM and IP-based calls.

One voice resource per channel

This mode is entered if voice resources are licensed but none of the previously described modes can be selected. This mode allows to play or to record but not at the same time. If a recording is active, any *DivaSendVoice* function will fail.

When a call is established, the received audio is not automatically signaled to the application via the event *DivaEventDataAvailable*. If the application requires this event, it must enable this via the function *DivaEnableRxData* and ensure that no play is active.

Conference resources

The Dialogic® Diva® SDK uses the HMP conference resources when an IP-based call is added to a conference. Note that creating a conference via *DivaCreateConference* will always succeed, even if no conference resource is licensed. The conference resource is allocated when an IP-based call is added to a conference using *DivaAddToConference*.

For each IP-based call that is added to a conference, one conference party resource is allocated and released when the call is disconnected or removed from the conference. An additional conference party resource is required if TDM and IP calls are bridged or if a play or record operation on the conference object is initiated. Note that once this additional resource is allocated, it remains at the conference object until the last IP-based call is removed from the conference or the conference object is released using *DivaDestroyConference*.

Applications may record from any conference member. Playing to an IP-based call that is part of a conference is not possible. In this case, the function *DivaSendVoiceFile* and the other sent voice-related functions will return *DivaErrorInvalidState*. The same is valid for a call that is line interconnected (tromboned) to another call.

Fax resources

The fax resources licensed for the Dialogic® HMP software support T.38 and clear channel fax, and the maximum supported speed is 14.400 bps. Fax resources are allocated when the application initiates a fax call or when the remote peer indicates a call as a fax call. For Dialogic HMP-based IP calls, the supported fax data format is TIFF. The following fax formats are supported:

- DivaFaxFormatTIFF_G3
- DivaFaxFormatTIFF_G4
- DivaFaxFormatColorJPEG

CHAPTER 3

Dialogic® Diva® Management API Overview

About the Dialogic® Diva® Management API

The Diva Management API is used to access (read and write) internally available information provided by the underlying Diva communication platform. In addition, the Diva Management API can be used to execute specific actions or commands, which in turn enable the client to do call control and monitoring and to read D-channel and B-channel statistics.

The internally available information can be divided into the following categories:

- Version information about the firmware and DSP code
- Configuration information (configured ISDN numbers, switch type, etc.)
- Statistics information (number of packets sent or received, error counters, etc.)
- Status information (state values, number of active calls, etc.)

Standard IDI mechanisms (REQUESTs, INDICATIONs, information elements) are defined to access management information. Due to these mechanisms, a generic tool can be used to handle the management information structure.

The Management Interface is structured like a virtual file space. It contains nodes (similar to directories and values (similar to files). Each node or value is identified by its path and name. A path contains node names followed by a specific value or node (similar to a subdirectory) that is separated by a backslash "\". Each node can contain further nodes or values.

Prerequisites

This manual assumes the developer has knowledge of Microsoft® Visual C++® 6.0 or higher.

CHAPTER 4

Functions

The management information can either be accessed by the IDI or via a high-level interface provided by the Dialogic® Diva® Management API. The high-level functions are made available by the Diva Management API library, DIMGMTIF.DLL that is provided with the Dialogic® Diva® boards and is also available in the Dialogic® Diva® SDK. The access is done by a simple C call interface. The requested information is specified by indicating node and name. The Diva SDK provides these names in the header file DivaMgmtIf.h, which also includes function prototypes. A library for external references is also available. The currently defined C call interface provides the following functions:

Diva_Mgmt_GetVersion

This function returns the current version number of the Dialogic® Diva® Management API.

```
DWORD        Diva_Mgmt_GetVersion (    )
```

Parameters

None.

Return values

The return value is the version number of the Diva Management API. The lower byte of the low word contains the minor version number. The higher byte of the low word contains the major version. The upper word is reserved. The version number 1.00 would be presented as 0x00000100.

Remarks

The application should check the version number to ensure that it was built for this type of interface. The version number will only change if the interface itself changes, not if the API changes internally. Note that the Dialogic® Diva® SDK header files contain a definition for the version number (DIVA_MGMT_IF_VERSION).

Diva_Mgmt_RegisterInstance

This function is issued to register an application with the Diva Management API.

```
DWORD     Diva_Mgmt_RegisterInstance (  DWORD     dwVersion,
                                        HANDLE    *phInstance );
```

Parameters

dwVersion

[in] Version number of Diva Management Interface used when the application was built.

phInstance

[out] Pointer to a value that receives the instance identifier.

Return values

The function returns IDI_SUCCESS if the registration succeeded. Potential error codes are ERROR_IDI_NO_MEMORY or ERROR_IDI_ILLEGAL_VERSION.

Remarks

An application must register with the Dialogic® Diva® Management API before it can use any other function. The instance identifier assigned during registration is needed in other calls, e.g., for getting a statistics object.

When the first application registers, the access to the Dialogic® Diva® boards is initiated and the communication channels are created.

Diva_Mgmt_ReleaseInstance

This function releases all resources that belong to the application instance.

```
DWORD    Diva_Mgmt_ReleaseInstance (  HANDLE    hInstance );
```

Parameters

hInstance

[in] Defines the instance to release. This is the value assigned during registration.

Return Values

The function returns IDI_SUCCESS if the instance is successfully released. It returns ERROR_IDI_INVALID_HANDLE if the instance is unknown.

Remarks

A call to Diva_Mgmt_ReleaseInstance frees the resources allocated for this instance. The pointers returned for any requested interface of any board are no longer valid. If this is the last release for the last registered interface, the access to the board(s) will also be terminated.

Diva_Mgmt_EnumAdapter

This function provides information about installed Dialogic® Diva® boards.

```
DWORD    Diva_Mgmt_EnumAdapter (   HANDLE               hInstance,
                                   HANDLE               *hEnum,
                                   DIVA_ADAPTER_INFO    *pInfo );
```

Parameters

hInstance

[in] Identifies the application instance. This instance has been assigned by a previous call to Diva_Mgmt_RegisterInstance.

hEnum

[in] [out] Used internally by the enumeration process. Must be set to zero for the first call.

Info

[out] Pointer to a value of the type DIVA_ADAPTER_INFO that receives the information for the enumerated board.

Return values

The function returns IDI_SUCCESS if *pInfo* contains information about a board. It returns ERROR_IDI_INVALID_PARAMS if the parameters are invalid, e.g. *pInfo* is not set. It returns ERROR_IDI_NO_MORE_ADAPTER if no more board information is available.

Remarks

An application calls Diva_Mgmt_EnumAdapter to retrieve information about the installed board. Note that Dialogic® Diva® 2PRI and Analog-2 Media Boards represent two adapters because of their two controllers. Dialogic® Diva® 4BRI-8M, 4PRI, and Analog-4 Media Boards represent four boards because of their four controllers.

The function returns the information in a structure of the type DIVA_ADAPTER_INFO which contains the following information:

Parameter name	Description
hAdapter	Handle of board to be used for any board-specific function, e.g. retrieve the interface pointer for a board
Name	Product name of the board
dwSerialNumber	Serial number of the board
dwAdapterPerCard	Number of lines (each line represents a board) per card
dwAdapterIndex	Line (board) index starting with zero (if more than one line is available)
dwChannels	Number of channels supported by this board
bModemSupport	If true, the board supports modem connections
bFaxSupport	If true, the board supports fax group 3 connections
bVoIPSupport	If true, the board supports RTP streaming and VoIP codecs
bAnalogBased	If true, the board is a Diva Analog Media Board.

Diva_Mgmt_GetValue

Diva_Mgmt_GetValue is the central function of the Dialogic® Diva® Management API. It retrieves information from the nodes and values of the Diva Management API.

```
DWORD    Diva_Mgmt_GetValue (    HANDLE          hInstance,
                                 HANDLE          hAdapter,
                                 char            *pValueName,
                                 void            *pBuffer,
                                 DWORD           DataLength,
                                 DWORD           *pBytesRead );
```

Parameters

hInstance

[in] Instance assigned to the application with Diva_Mgmt_RegisterInstance (see Diva_Mgmt_RegisterInstance on page 13).

hAdapter

[in] Identifies the Dialogic® Diva® board. The handle has been returned during board enumeration with Diva_Mgmt_EnumAdapter. If only one board is installed, this can be set to zero (see Diva_Mgmt_EnumAdapter on page 14).

pValueName

[in] Identifies which information should be retrieved. This is a zero-terminated string containing the path and name of the value that should be read, e.g. State\Layer1.

pBuffer

[out] Pointer to a memory location where the data should be placed.

DataLength

[in] Length in bytes of the memory location to which *pBuffer* points.

pBytesRead

[out] Points to a memory location that receives the number of bytes written to *pBuffer*. The caller may set this value to NULL if this information is not required.

Return values

The function returns IDI_SUCCESS if the information has been placed in the buffer. If more than one board is installed but no board has been specified, ERROR_IDI_NO_MEMORY is returned.

Remarks

An application uses this function to retrieve any management data. The parameters are specified in the tables starting with Config directory on page 18. The returned information, especially the data type, depends on the parameter and is also listed in the tables starting with Config directory on page 18.

CHAPTER 5

Nodes and Values

The Dialogic® Diva® Management API contains the following nodes and values. The values can be read by the Diva Management API functions. The node contents cannot be listed using the Diva Management API functions, but the node names are important for indicating correct paths to values.

A node can be read and returns the nodes and variables of the next deeper level of the directory tree. A variable can be read, written or marked as "Notify on change" depending on its attribute. The variable attributes are indicated as follows:

Variable attribute	Description
RO	Read only
W	Writable
EVENT	Variable can be used to generate event (notification)

Value types

Value types are primarily used to distinguish nodes from values and executable functions. The second function is to specify how a value has to be interpreted by the application or in which format it is intended to be displayed.

Value types are indicated as follows:

Value type	Description
MI_DIR	Directory
MI_HINT	Unsigned integer with hexadecimal representation, variable length, LSB first
MI_UINT	Unsigned decimal integer, variable length, LSB first
MI_ASCIIZ	ASCII string, zero terminated
MI_ASCII	ASCII string, first byte is length
MI_BOOLEAN	Boolean value, TRUE or FALSE
MI_BITFLD	Unsigned integer, representation as bit field (sequence of 0/1's), variable length
MI_EXECUTE	Executable function
MI_NUMBER	Plain hexadecimal, first byte is length
MI_HSTR	Byte sequence represented as non-separated two-digit hex values
MI_IP_ADDRESS	Dot-separated decimal number
MI_SPID_STATE	Number representing the D-channel link state

Root directory

The root directory of the Dialogic® Diva® Management API contains the following values and subdirectories:

Name	Type	Attribute	Comment
MIF Version	MI_HINT	RO	This value displays the version number of the management interface.
Build	MI_ASCIIZ	RO	This value displays the name and build number of the used protocol file.
Events down	MI_UINT	EVENT	When a board is in overload condition, the management interface's event tracing is paused for each variable until internal resources are re-available.
Config	MI_DIR		A directory that contains state machines, selected protocols and their parameters.

Name	Type	Attribute	Comment
Statistics	MI_DIR		A directory that contains information on the number of sent or received calls and packets, error counters, etc.
State	MI_DIR		A directory that contains complete information on the status of calls and the layer 1 and layer 2 status.
Trace	MI_DIR		A directory that contains trace functions (every application instance will receive an individual copy of the trace information).

Config directory

The "Config" directory contains ISDN values and parameters of your Dialogic® Diva® board. You can use the values in this directory and its sub-directories to view and modify your software configuration.

Name	Type	Attribute	Comment
US-EKTS	MI_DIR		Dialogic® Diva® PRI and 4BRI only. This subdirectory includes the values related to switch-specific supplementary services messages for North American protocols.
DChannel	MI_DIR		Diva PRI only. This subdirectory includes all values related to the D-channel protocol (switch type). Go to this directory if you need to review or modify your D-channel protocol settings.
Layer 2	MI_DIR		Diva PRI only. This subdirectory includes all values related to the layer 2 activation policy. Go to this directory if you need to review layer 2 parameters.
Layer 1	MI_DIR		Diva PRI only. Go to this subdirectory to review or modify the layer 1 framing type.
Modem	MI_DIR		This subdirectory contains information on analog modem parameters. It is used for internal debug options only.
Fax	MI_DIR		This subdirectory contains information on fax parameters. It is used for internal debug options only.
Robbed Bit	MI_DIR		Diva PRI only. This subdirectory includes all values related to the Robbed Bit Signaling switch type.
Version	MI_UINT	RO	This value indicates the protocol version.
DID-Length	MI_UINT	RO	Diva PRI only. When this value is set, the incoming digits of the called number are collected and the call is not forwarded to the application until their length corresponds to the given DID-Length. **Note:** Please see legal notice at the front of this document.
Redial-Timeout	MI_UINT	RO	Diva PRI only. This value indicates the time in seconds within which only a restricted number of redial attempts is possible. The number of possible attempts is defined by the ‘Redial-Attempts’ value. After the ‘Redial-Timeout’ has expired, new redial attempts may be made.
Redial-Attempts	MI_UINT	RO	Diva PRI only. This value indicates the number of redial attempts that may be made within the period defined by the ‘Redial-Timeout’ value.
QSIGDialect	MI_UINT	RO	This variable indicates which QSIG version is used. Possible return values are: • 0: Autodetection • 1: ECMA-QSIG • 2: ETSI-SS • 3: ISOV2-QSIG • 4: CornetNQ

Name	Type	Attribute	Comment
Ringertone	MI_BOOLEAN	RO	Dialogic® Diva® PRI only. This value indicates if local tones (busy, alert) are to be played to the network when no tones are provided by the network. The returned values are as follows: • True: Local tones are generated. • False: No local tones are generated.
NT	MI_BOOLEAN	RO	This value indicates if the installed board is operated in NT or TE operation mode. Can be used only for NT-capable boards. In NT (Network Termination) operation mode the board emulates a switch. The returned values are as follows: • True: Board is operated in NT mode. • False: Board is operated in TE mode.
PRI	MI_BOOLEAN	RO	This value indicates whether a Dialogic® Diva® PRI or BRI board is installed. The returned values are interpreted as follows: • True: A PRI board is installed. • False: A BRI board is installed.
GlobalCRs	MI_BOOLEAN	RO	This value turns processing of D-channel messages with global call reference on or off. The returned values are interpreted as follows: • True: D-channel messages with global call reference is turned on. • False: D-channel messages with global call reference is turned off.
NoOrderCheck	MI_BOOLEAN	RO	This value turns the order check of information elements in D-channel messages on or off. The returned values are interpreted as follows: • True: Order check of information elements is turned off. • False: Order check of information elements is turned on.
NT-2	MI_BOOLEAN	RO	This value turns NT 2 (Direct Inward Dialing) mode on or off. • True: NT2 mode is turned on. • False: NT2 mode is turned off.
LoopBack	MI_BOOLEAN	RO	This value turns the loop back on the physical interface on or off. • True: Loop back is turned on. • False: Loop back is turned off.
CardType	MI_UINT	RO	Indicates the Diva board type.
DSPState	MI_BITFLD	RO	A bit field to identify the presence of active DSPs. Each bit stands for a DSP. If the bit is set, a DSP is present.
SerialNumber	MI_UINT	RO	Indicates the serial number of the board. It is automatically read from the board.
BoardRevision	MI_UINT	RO	Indicates the hardware revision of the board.
SubFunction	MI_UINT	RO	Sub-function.
SPID-#	MI_DIR		A subdirectory that includes all D-channel link configuration parameters.

Config/US-EKTS subdirectory

In the "Config/US-EKTS" subdirectory, you can get information on the switch-specific supplementary services messages for the North American switch types.

Note: This subdirectory is available only for Dialogic® Diva® PRI and 4BRI boards.

Name	Type	Attribute	Comment
CACH Appearances	MI_UINT	RO	This value indicates the number of call appearances (call references) from which on CACH (Call Appearance Call Handling) needs to be activated.
Key BeginConf	MI_HINT	RO	Indicates the key value that activates the begin conference feature.
Key DropConf	MI_HINT	q	Indicates the key value that activates the drop conference feature.
Key CallTransfer	MI_HINT	RO	Indicates the key value that activates the call transfer feature.

Config\DChannel subdirectory

In the "Config\Dchannel" subdirectory, you can get information on the currently active switch type and set the switch type that you want to use.

Note: This subdirectory is only available for Diva PRI boards.

Name	Type	Attribute	Comment
Name	MI_ASCIIZ	RO	Indicates the name of the currently active D-channel protocol.
Protocol names that can be selected	MI_EXECUTE		Execute this function to select appropriate protocol
Protocol	MI_UINT	W	Active protocol number; writing a number to this variable will select the appropriate protocol
Release	MI_ASCIIZ	RO	Indicates the build time and date of the protocol image.

Config\Layer2 subdirectory

In the subdirectory "Config\Layer2", you can get information on the layer 2 activation strategy and modify it, if required.

Note: This subdirectory is only available for Diva PRI boards.

Name	Type	Attribute	Comment
Stable L2	MI_ASCIIZ	RO	Indicates the name of the currently active layer 2 activation strategy. The following return values are possible: • On demand: Layer 2 is disconnected after last active connection was cleared. • Send no disc: Layer 2 is disconnected only if requested by the NT side. • Permanent: Layer 2 is kept always active.
On demand (0)	MI_EXECUTE		Selects "on demand" activation strategy, i.e. layer 2 is disconnected after last active connection was cleared
Send No Disc (1)	MI_EXECUTE		Selects "no disc" activation strategy, i.e. layer 2 is disconnected only if requested by the NT side

Name	Type	Attribute	Comment
Permanent (2)	MI_EXECUTE		Selects "permanent" activation strategy, i.e. board will try to keep layer 2 always active
L2	MI_UINT	RO	Indicates the number of the currently active layer 2 activation strategy. Writing to this variable will change the layer 2 activation strategy without reset of the D-channel protocol state, i.e. the state will be changed after last active connection was cleared. The various activation strategies are assigned to the following numbers: 0: On demand 1: Send No Disc 2: Permanent

Config\Layer1 subdirectory

In the "Config\Layer1" subdirectory, you can get information on the currently used layer 1 framing type and modify it if required.

Note: This subdirectory is only available for Diva PRI boards.

Name	Type	Attribute	Comment
Current name	MI_ASCIIZ	RO	Indicates the name of the currently used layer 1 framing type. The following return values are possible: • National default: The CRC4 information element is handled as defined in the specification of the selected D-channel protocol. • No CRC4: CRC4 is not transmitted and checked in incoming packets. • CRC4: CRC4 is always transmitted and checked in incoming packets. • Autodetection: Automatic detection of how CRC4 is to be handled.
Current	MI_UINT	RO	Indicates the number of the currently used layer 1 framing type, can be written to select a new layer 1 framing type. The various framing types are assigned to numbers as follows: 0 - National default 1 - Double framing (No CRC4) 2 - Multiframing (CRC4) 3 - Autodetection
Auto CRC4	MI_EXECUTE	RO	Selects auto detection of layer 1 framing type, i.e. the CRC 4 information element is handled as specified in the ISDN protocol’s specification.
CRC4	MI_EXECUTE	RO	Selects CRC4. The CRC 4 information element is always transmitted.
No CRC4	MI_EXECUTE	RO	Selects double framing. No CRC 4 information element is transmitted.

Config\Robbed Bit subdirectory

In the "Config\Robbed Bit" subdirectory, you can get information on the parameters used for robbed bit signaling.

Note: This subdirectory is only available for Diva PRI boards.

Name	Type	Attribute	Comment
Options	MI_BITFLD	RO	This bitfield represents the settings for the following parameters (starting from the two least significant bits): • Glare-resolving party -Indicates how a call collision is resolved: 00 - No: Your server waits for the remote side to return to on-hook status and then starts sending dial digits. 01 - Yes: Your server returns to on-hook status and waits for dial digits from the remote site. • Trunk type - Indicates which trunk mode is used: 00 - Wink start 01 - Loop start 10 - Ground start • Dial type - Indicates which dial type is used: 00 - Pulse dialing 01 - DTMF (Dual Tone Multifrequency) 10 - MF (Multifrequency) • Direct Inward Dialing (pulse dialing only): Indicates if incoming dialed digits are collected and forwarded to the application or if the call is accepted as soon as the destination party indicates off-hook status. **Note:** Please see legal notice at the front of this document.
Answerdelay	MI_UINT	RO	Indicates the time (in seconds) after which the board hangs up if the remote station does not answer a call.
Digtitimeout	MI_UINT	RO	Indicates the time (in seconds) that may pass between incoming dialed digits. If this time interval is exceeded, the called number is assumed to be complete and the call is forwarded to the application.
Bearer Cap	MI_UINT	RO	Indicates the bearer capability that is to be included with the incoming call that is signaled to the application. Possible return values are: • 4 - Analog/voice (3.1 kHz) • 8 - Digital/data
Debug	BITFLD	RO	Indicates the debug mask that is to be used for robbed bit signaling.

Config\SPID-# subdirectory

In the "Config\SPID-#" subdirectory, you can get information on the D-channel link configuration and modify it if required.

Name	Type	Attribute	Comment
Automatic TEI	MI_BOOLEAN	RO	Indicates that automatic TEI assignment is enabled. Automatic TEI assignment is used if the board is operated in a point-to-multipoint environment.
TEI	MI_UINT	RO	Indicates that a fixed TEI value is assigned to the connection. Fixed TEIs are used when the interface type is a point-to-point connection or a leased line.
SPID	MI_ASCII	RO	Indicates the SPID number string. The SPID number is only required for the North American switch types NI-1 and AT&T if they are used with a basic rate interface.
Number	MI_ASCII	RO	Indicates the origination address.
Sub-Adr	MI_ASCII	RO	Indicates the origination sub-address.

Statistics directory

The "Statistics" subdirectory contains detailed information on the number of outgoing and incoming calls, the reasons for disconnection, the bytes and frames sent and received by the D-channel and the B-channels, and the charges.

Name	Type	Attribute	Comment
Outgoing Calls	MI_DIR		Indicates statistic information for outgoing calls.
Incoming Calls	MI_DIR		Indicates statistic information for incoming calls.
D-Layer2	MI_DIR		Indicates the sum of D-channel layer 2 statistics.
B-Layer2	MI_DIR		Indicates the sum of B-channel layer 2 statistics.
Charges	MI_UINT	RO	Indicates the sum of charge units.
Clear Values	MI_EXECUTE		Resets all values in this directory and all subdirectories.

Statistics\Outgoing Calls subdirectory

In the "Statistics\Outgoing Calls" subdirectory, you can get information on how many outgoing calls have been attempted and established and the reasons for rejection of calls.

Name	Type	Attribute	Comment
Calls	MI_UINT	RO	Indicates the number of requested calls.
Connected	MI_UINT	RO	Indicates the number of successfully established calls.
User busy	MI_UINT	RO	Indicates the number of calls that were rejected because the user was busy.
No Answer	MI_UINT	RO	Indicates the number calls that were rejected because the remote station did not answer the call.
Wrong Number	MI_UINT	RO	Indicates the number of calls that were rejected because the wrong number was dialed.
Call rejected	MI_UINT	RO	Indicates the number of calls that were refused because the remote station rejected them.
Other failures	MI_UINT	RO	Indicates the number of calls that were rejected because of other reasons than the ones described above.
Clear values	MI_EXECUTE		Resets all values in this directory.

Statistics\Incoming Calls subdirectory

In the "Statistics\Incoming Calls" subdirectory, you can get information on how many incoming calls have been attempted and accepted and the reasons for rejection of calls.

Name	Type	Attribute	Comment
Calls	MI_UINT	RO	Indicates the number of incoming calls.
Connected	MI_UINT	RO	Indicates the number of accepted calls.
User busy	MI_UINT	RO	Indicates the number of calls that were rejected because the user was busy.
Call rejected	MI_UINT	RO	Indicates the number of calls that were refused because the called party rejected them.
Wrong number	MI_UINT	RO	Indicates the number of calls that were rejected because the wrong number was dialed.
Incompatible destination	MI_UINT	RO	Indicates the number of calls that were rejected because the destination was not compatible with the TE.
Out of order	MI_UINT	RO	Indicates the number of calls that were rejected because the destination was out of order.

Name	Type	Attribute	Comment
Ignored	MI_UINT	RO	Indicates the number of calls that were rejected because the call was ignored.
Clear values	MI_EXECUTE		Resets all values above.

Statistics\D-Layer2 and Statistics/B-Layer2 subdirectories

The "Statistics D-Layer2" and "Statistics/B-Layer2" subdirectories contain identical values. In these directories, you can get statistic information on the frames and bytes that were sent and received in the D-channel and all B-channels. You can also retrieve information on the number of transmit or receive errors.

Name	Type	Attribute	Comment
X-Frames	MI_UINT	RO	Indicates the number of frames sent.
X-Bytes	MI_UINT	RO	Indicates the number of bytes sent.
X-Errors	MI_UINT	RO	Indicates the number of detected transmit errors.
R-Frames	MI_UINT	RO	Indicates the number of received frames.
R-Bytes	MI_UINT	RO	Indicates the number of received bytes
R-Errors	MI_UINT	RO	Indicates the number of detected receive errors.

State directory

The "State" directory contains all important information required to monitor your connections.

Name	Type	Attribute	Comment
Layer1	MI_ASCIIZ	RO	Indicates the state of layer 1: activated or down.
Layer2 No#	MI_SPID_STATE	RO	Indicates the state of the D-channel link number #. The state is assigned to the following values: 0: Idle 1: Layer2 up 2: Layer2 disconnecting 3: Layer2 connecting 4: SPID initializing (for North American protocols used with Dialogic® Diva® BRI boards only) 5: SPID initialized (for North American protocols used with BRI boards only)
CODEC #	MI_DIR		Adjustment and states of CODEC number # (this directory is present only with CTI-capable boards equipped with audio codec).
B#	MI_DIR		Indicates the state of B-channel number #. Each B-channel has its own directory.

Path:\State\CODEC # (Diva CT board only)

Name	Type	Attribute	Comment
Mic Volume Hdset	MI_UINT	W	Volume of head/handset microphone 1 Range: 0...8
Spk Volume HdSet	MI_UINT	W	Volume of head/handset speaker 1 Range: 0...9
Mic Volume HFree	MI_UINT	W	Volume of hands-free (ext) microphone Range: 0...8
Spk Volume HFree	MI_UINT	W	Volume of speaker in hands-free mode Range: 0...12
Spk Volume LoudH	MI_UINT	W	Volume of speaker in loudhearing mode Range: 0...12
Transmit Volume	MI_UINT	W	Gain adjustment of transmit path1(micro) Range: 0...51
Receiver Volume	MI_UINT	W	Gain adjustment of receive path1(spk) Range: 0...51
Sidetone Gain	MI_UINT	W	Hand/Headset 1 sidetone gain adjustment Range: 0...51
Ring Volume Ext	MI_UINT	W	Volume of external ringer (ext. speaker) Range: 0...12
Ring Melody	MI_UINT	W	Selection of predefined ringer melodies Range: 0...9
IntrTone Volume	MI_UINT	W	Volume 1 of intrusion tone melody Range: 0...10
IntrTone Melody	MI_UINT	W	Selection of actual intrusion tone melody Range: 0...9
Muting on/off	MI_UINT	W	Switches the microphone 1 off Range: 0...1
HdSet 1 Status	MI_ASCIIZ	EVENT	Plug state / hook state of hand/headset 1 Strings: 'Unplugged', 'Operating', 'Hook ON'. A handset can be Operating or Hook ON, a headset can be Operating or Unplugged
Ext Mic Status	MI_ASCIIZ	EVENT	Plug state of the external microphone Strings: 'Unplugged', 'Operating'
HdSet 2 Status	MI_ASCIIZ	EVENT	Plug state / hook state of hand/headset 2 Strings see HdSet1 Status
Switch Status	MI_ASCIIZ	EVENT	Telephone chip (codec) operating mode Strings: 'Off', 'Head/HandSet ON', 'External Mic/Spk ON', 'Alerting External Speaker', 'Alerting Head/Handset'; the Alerting Head/Handset is active if an intrusion tone is used to ring a call
Switch Hdset/Ext	MI_EXECUTE		Toggles between hands-free and headset
Mic Vol 2 HdSet	MI_UINT	W	Volume of head/handset microphone 2 Range: 0...8
Spk Vol 2 HdSet	MI_UINT	W	Volume of head/handset speaker 2 Range: 0...9
Transmit Vol 2	MI_UINT	W	Gain adjustment of transmit path2 (micro) Range: 0...51
Receiver Vol 2	MI_UINT	W	Gain adjustment of receive path2 (spk) Range 0...51
Sidetone Gain 2	MI_UINT	W	Hand/Headset 2 sidetone gain adjustment Range: 0...51

Name	Type	Attribute	Comment
IntrTone Vol 2	MI_UINT	W	Volume 2 of intrusion tone melody Range 0...10
Muting 2 on/off	MI_UINT	W	Switches microphone 2 off Range: 0...1

State\B#

This directory contains the currently used parameters for the respective B-channel number. If the B-channel is inactive, the parameters used for the last call are displayed.

Name	Type	Attribute	Comment
Framing	MI_ASCIIZ	RO	Indicates the currently used layer 1 protocol. The following return values are possible: • Idle • HDLC 64K • Codec • V.110 HDLC • V.110 Async • V.110 CM • Fax • Modem Async • Modem HDLC • Modem Async-HDLC • V.110 Async-HDLC • Voice • HDLC 56K • Transparent
Line	MI_ASCIIZ	RO	Indicates the line state. It can return the following values: • Idle • Connected-in • Connected-out
Layer2	MI_ASCIIZ	RO	Indicates the currently used layer 2 protocol. The layer 2 parameter returns one of the following values: • X.75 • Transparent • SDLC • LAPD • LAPB • V.120 • V.42 • AT Parser • X.75/V.42bis • RTP • V.120 • PIAFS
Layer3	MI_ASCIIZ	RO	Indicates the currently used layer 3 protocol. The layer 3 parameter returns one of the following values: • T.70NL • X.25PLP • T.70NLX • Transparent • ISO8208 • T.30

Name	Type	Attribute	Comment
Line type	MI_ASCIIZ	RO	Indicates the currently used line type. The line type parameter can have the following values: • DialUp: The connection is established on demand and held until no longer requested. • Leased: Two stations are connected permanently.
Remote Address	MI_ASCII	RO	Indicates the address of the remote station. For outgoing calls, the remote address is the called party number. For incoming calls, it is the calling party number.
Remote Subaddr	MI_ASCII	RO	Indicates the subaddress of the remote station. For outgoing calls, the remote subaddress is the subaddress of the called party. For incoming calls, it is the subaddress of the calling party.
Local Address	MI_ASCII	RO	Indicates the address of the local station. For outgoing calls, the local address is the calling party number. For incoming calls, it is the called party number.
Local Subaddr	MI_ASCII	RO	Indicates the subaddress of the local station. For outgoing calls, the local subaddress is the subaddress of the calling party. For incoming calls, it is the subaddress of the called party.
BC	MI_NUMBER	RO	Indicates the currently used bearer capability, e.g. 88 90 for unrestricted 64K digital data calls.
HLC	MI_NUMBER	RO	Indicates the high layer compatibility.
LLC	MI_NUMBER	RO	Indicates the low layer compatibility.
Charges	MI_UINT	RO	Indicates the sum of line-related charges.
Call Reference	MI_HINT	RO	Indicates the call reference number in the D-channel.
Last Disc Cause	MI_HINT	RO	Indicates the last disconnect cause, e.g. user busy, if available.
User ID	MI_ASCII	RO	Indicates the last owner that claimed this channel, if any.
Clear Call	MI_EXECUTE		Sends DISCONNECT for channel B# if, for example, the call duration exceeds a fixed limit.

Trace directory

The "Trace" directory contains all parameters that you can use to adjust the tracing to your individual requirements.

Name	Type	Attribute	Comment
B-Ch# Enable	MI_BITFLD	W	This variable is a bit field that defines the B-channels to be traced. Default is all channels. A channel is traced if its corresponding bit is set to 1. The least significant bit (LSB) is assigned channel 1, the others follow in ascending order.
Debug Level	MI_HINT	W	This variable selects debug trace events to be traced or not. Default is all debug trace events. All debug trace events below this debug Level are traced and all debug trace events equal or greater debug level are ignored. A debug level of 0 disables the debug trace events. The debug levels are assigned numbers as follows:
D-HW Txt Mask	MI_BITFLD	W	D-channel hardware trace mask. The Txt Mask selects a text trace from a module to be traced or not. A text trace is generated in the source with its according mask bit(s) set. This mask is logically ANDed with the module Txt Mask. If the value is not equal to 0, the text is traced, else it is suppressed.
B-HW Txt Mask	MI_BITFLD	W	B-channel hardware trace mask
Misc Txt Mask	MI_BITFLD	W	Trace mask for miscellaneous use

<table>
<tr><th>Name</th><th>Type</th><th>Attribute</th><th>Comment</th></tr>
<tr><td>Event Enable</td><td>MI_BITFLD</td><td>W</td><td>Bit field that defines the group of event types to be traced. Events are traced if the corresponding bit is set. Default is bit 0-6 set (corresponding events enabled). The event groups are assigned to bit positions as follows:
<table>
<tr><th>Event Group</th><th>Log symbol</th><th>Bit position</th></tr>
<tr><td>D-channel</td><td>D-...</td><td>Bit 0 (LSB)</td></tr>
<tr><td>Low layer</td><td>LL...</td><td>Bit 1</td></tr>
<tr><td>Network layer</td><td>N-...</td><td>Bit 2</td></tr>
<tr><td>Data link error</td><td>MDL...</td><td>Bit 3</td></tr>
<tr><td>Layer 1</td><td>-</td><td>Bit 4</td></tr>
<tr><td>Call comment</td><td>SIG...</td><td>Bit 5</td></tr>
<tr><td>Modulation data</td><td>EYE...</td><td>Bit 6</td></tr>
<tr><td>All other events</td><td>-</td><td>Bit 15 (MSB)</td></tr>
</table></td></tr>
<tr><td>Max Log Length</td><td>MI_UINT</td><td>W</td><td>This variable sets the maximum size in bytes of the data element inside the trace information element. This value is set to 30 bytes by default. Trace information elements belonging to the Call Comment Event Group are not limited by this variable.</td></tr>
</table>