Dialogic
Making Innovation Thrive™

Dialogic® Multimedia Kit Software Release 1.0 for PCIe

Release Update

October 31, 2008

Copyright and Legal Notice

Copyright © 2008, Dialogic Corporation. All Rights Reserved. You may not reproduce this document in whole or in part without permission in writing from Dialogic Corporation at the address provided below.

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 EXPLICITLY SET FORTH BELOW OR 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.

Due to differing national regulations and approval requirements, certain Dialogic products may be suitable for use only in specific countries, and thus may not function properly in other countries. You are responsible for ensuring that your use of such products occurs only in the countries where such use is suitable. For information on specific products, contact Dialogic Corporation at the address indicated below or on the web at www.dialogic.com.

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. **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, Dialogic Pro, Brooktrout, Cantata, SnowShore, Eicon, Eicon Networks, Eiconcard, Diva, SIPcontrol, Diva ISDN, TruFax, Realblocs, Realcomm 100, NetAccess, Instant ISDN, TRXStream, Exnet, Exnet Connect, EXS, ExchangePlus VSE, Switchkit, N20, Powering The Service-Ready Network, Vantage, Connecting People to Information, Connecting to Growth, Making Innovation Thrive, and Shiva, among others as well as related logos, are either registered trademarks or trademarks of Dialogic. 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.

Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries. The other names of actual companies and products mentioned herein are the trademarks of their respective owners.

Using the AMR-NB resource in connection with one or more Dialogic products mentioned herein does not grant the right to practice the AMR-NB standard. To seek a patent license agreement to practice the standard, contact the VoiceAge Corporation at www.voiceage.com/licensing.php.

Publication Date: October 31, 2008

Document Number: 05-2636-002

About This Publication

This section contains information about the following topics:

- Purpose
- Intended Audience
- How to Use This Publication
- Related Information

Purpose

This Release Update addresses issues associated with the Dialogic® Multimedia Kit Software Release 1.0 for PCIe. In addition to summarizing issues that were known as of this release, it is intended that the Release Update will continue to be updated to serve as the primary mechanism for communicating any new issues that may arise after the release date.

Intended Audience

This Release Update is intended for all users of the Dialogic® Multimedia Kit Software Release 1.0 for PCIe.

How to Use This Publication

This Release Update is organized into the following sections (click the section name to jump to the corresponding section):

- Document Revision History: This section summarizes changes and additions that have been made to this Release Update after its original release. This section is organized by document revision and document section.
- Post-Release Developments: This section describes significant changes to the release subsequent to the general availability release date. For example, new features provided in the Service Update are described in this section.
- Release Issues: This section lists issues that may affect the system release hardware and software.
- Documentation Updates: This section contains corrections and other changes that apply to the documentation not made prior to the release. These updates are organized by documentation category and by individual document.

Related Information

See the following for additional information:

- For information about the products and features supported in this release, see the *Dialogic® Multimedia Kit Software Release 1.0 for PCIe Release Guide*, which is included as part of the documentation bookshelf for the release.
- For further information on issues that have an associated defect number, you may use the Defect Tracking tool at *http://membersresource.dialogic.com/defects/*. When you select this link, you will be asked to either LOGIN or JOIN.
- *http://www.dialogic.com/manuals/* (for Dialogic® product documentation)
- *http://www.dialogic.com/support/* (for Dialogic technical support)
- *http://www.dialogic.com/* (for Dialogic® product information)

Document Revision History

This Revision History summarizes the changes made in this and each previously published version of the Release Update for Dialogic® Multimedia Kit Software Release 1.0 for PCIe, which is a document that is planned to be periodically updated throughout the lifetime of the release.

Document Rev 02 - published October 31, 2008

Updated for Service Update 107.

In the Post-Release Developments chapter:

- Added Service Update.
- Added MONA Support in the 3G-324M API.
- Added Capturing a Still Image from a Video Stream.
- Added Digital Video Recorder (DVR) Controls.
- Added Support for SS7 Functionality.
- Added Support for Dialogic® SS7 Boards.
- Added RTCP Receiver Report Enhancements.
- Added Multiple Frames Per Packet Support for AMR-NB.
- Added AMR-NB and G.711 Audio Over Nb UP.
- Added H.263 Using RFC 2429 (RFC 4629) Packetization.

In the Release Issues chapter:

- Added the following Known Issues: IPY00078401, IPY00078650, IPY00078715.
- Added the following Resolved Issues: IPY00043272, IPY00044505, IPY00044655, IPY00044656, IPY00044824, IPY00044834, IPY00045305, IPY00045522, IPY00078621.

In the Documentation Updates chapter:

- Added that a new version of Dialogic® Multimedia Kit Software Release 1.0 for PCIe Release Guide is now available on the documentation bookshelf.
- Added that a new version of Dialogic® 3G-324M API Programming Guide and Library Reference is now available on the documentation bookshelf.
- Added that a new document named Dialogic® Global Call SS7 Technology Guide is now available on the documentation bookshelf.
- Added updates to Dialogic® IP Media Library API Library Reference for IPM_NBUP_PROFILE_INFO, IPM_NBUP_SUBFLOW_INFO, and IPM_VIDEO_CODER_INFO data structures.

- Added that a new version of Dialogic® Device Management API Library Reference is now available on the documentation bookshelf.
- Added that a new version of Dialogic® Multimedia API Programming Guide and Library Reference is now available on the documentation bookshelf.
- Added that a new version of Dialogic® Standard Runtime Library API Library Reference is now available on the documentation bookshelf.

Document Rev 01 - published September 12, 2008

Initial version of document for Dialogic® Multimedia Kit Software Release 1.0 for PCIe.

Post-Release Developments 1

This section describes significant changes to Dialogic® Multimedia Kit Software Release 1.0 for PCIe subsequent to the general availability release.

1.1 Service Update

A Service Update for Dialogic® Multimedia Kit Software Release 1.0 for PCIe is now available. Service Updates provide fixes to known problems, and may also introduce new functionality. New versions of the Service Update will be released periodically. It is intended that this Release Update will document the features in the Service Updates.

1.2 MONA Support in the 3G-324M API

With Service Update 107, the MONA standard is supported in the 3G-324M API.

The Media Oriented Negotiation Acceleration (MONA) standard is a group of complementary procedures designed to significantly reduce delay in H.324 call setup time. The procedures include Media Preconfigured Channels (MPC), Accelerated Connect Procedure (ACP), and Signaling Preconfigured Channel (SPC). The MONA standard implemented by Dialogic uses MPC and ACP procedures, which classifies products based on Dialogic® 3G-324M software as Class II MONA terminals per H.324 Amendment K.7.2.1. Dialogic does not support the SPC procedure. For more information on MONA and its implementation in the 3G-324M API, see the *Dialogic® 3G-324M API Programming Guide and Library Reference*.

1.3 Capturing a Still Image from a Video Stream

With Service Update 107, the capture still image feature is supported. This feature provides the ability to capture a frame after a video stream has been paused and save it as an image. This feature is implemented in the Dialogic® Multimedia API Library. For more information, see the *Dialogic® Multimedia API Programming Guide and Library Reference*.

1.4 Digital Video Recorder (DVR) Controls

With Service Update 107, DVR controls are supported. These controls enable the user to pause, resume, and seek during video and audio playback. This feature is implemented in the Dialogic® Multimedia API Library. For more information, see the *Dialogic® Multimedia Programming Guide and API Library Reference*.

1.5 Support for SS7 Functionality

With Service Update 107, Dialogic® Multimedia Kit Software Release 1.0 for PCIe supports Signaling System 7 (SS7) via Signaling Interface Units (SIU) and SIGTRAN (IETF SS7 Signaling over IP). The Dialogic® system provides SIU/SIGTRAN detection, initialization and configuration support. The Dialogic® Global Call software supports the development of call control applications that use SS7 technology. For more information about SS7 technology, refer to the *Dialogic® Global Call SS7 Technology Guide*.

1.6 Support for Dialogic® SS7 Boards

With Service Update 107, Dialogic® Multimedia Kit Software Release 1.0 for PCIe supports Dialogic® SS7 boards. Dialogic® SS7 boards provide on-board support for SS7 common channel signaling protocols with a number of digital line interfaces (T1/E1/J1) and a H.100 or H.110 PCM highway that supports connection to a wide range of voice, data, and fax boards. See the *Release Guide* for supported Dialogic® SS7 boards. Refer to the *Dialogic® Global Call SS7 Technology Guide* for more details on the features and configurations.

1.7 RTCP Receiver Report Enhancements

Service Update 107 adds support for RTCP Receiver Report (RR) control packets for both audio and video streams. Previously only audio streams were supported.

For applications that stream multimedia via an RTSP (real-time streaming protocol) server, session participants must send RTCP RR control packets for both audio and video streams to report reception status and maintain session establishment. Failure to do so results in pre-mature session termination. This feature enables multimedia applications to interoperate with RTSP servers without unexpected session termination as a result of RTCP RR reports not being generated and sent to the RTSP server.

There are no API changes to the Dialogic® IP Media Library (IPML) API as a result of this feature.

1.8 Multiple Frames Per Packet Support for AMR-NB

With Service Update 107, multiple frames per packet (fpp) are supported for AMR-NB codec for RTP interface: 1 to 10 fpp (octet-aligned and bandwidth-efficient). Previously only 1 fpp was supported.

For more information, see the *Dialogic® IP Media Library (IPML) API Reference*.

Note: Using the AMR-NB resource in connection with one or more Dialogic products mentioned herein does not grant the right to practice the AMR-NB standard. To seek a patent license agreement to practice the standard, contact the VoiceAge Corporation at *http://www.voiceage.com/licensing.php*.

1.9 AMR-NB and G.711 Audio Over Nb UP

Service Update 107 adds support for streaming AMR-NB and G.711 audio over Nb UP. Previously only streaming 3G-324M over Nb UP was supported.

Note: Using the AMR-NB resource in connection with one or more Dialogic products mentioned herein does not grant the right to practice the AMR-NB standard. To seek a patent license agreement to practice the standard, contact the VoiceAge Corporation at *http://www.voiceage.com/licensing.php*.

1.9.1 Feature Description

This feature allows an IP media streaming (IPM) device to stream AMR-NB audio (20 ms, 12.2 Kbps bit rate) and G.711 audio (5 ms and 20 ms, A-law and mu-law) over Nb UP. This data can be streamed to and from a 3G network. A multimedia (MM) device can be connected to the IPM device for native play and record operations. Transcoding is not currently supported.

Setting up AMR-NB over Nb UP or G.711 over Nb UP media sessions is similar to setting up a 3G-324M over Nb UP session. However, the 3G-324M (M3G) component is not used. The following sections provide guidelines for streaming AMR-NB audio or G.711 audio over Nb UP.

1.9.2 Guidelines for Streaming AMR-NB Over Nb UP

To stream AMR-NB audio (20 ms, 12.2 Kbps bit rate) over Nb UP, follow these general guidelines:

- Specify a native connection between the IPM device and the MM device. Native play and record operation takes place using native AMR-NB 12.2 Kbps audio files.
- Where available, use the inline function to initialize a data structure. For example, use INIT_IPM_NBUP_PROFILE_INFO to initialize IPM_NBUP_PROFILE_INFO, and INIT_IPM_NBUP_INIT_SEND to initialize IPM_NBUP_INIT_SEND.
- Call **ipm_StartMedia( )** to start the session. In the IPM_NBUP_PROFILE_INFO structure, specify NBUP_PROFILE_AMR_NB in the eProfileType field.
- Call **ipm_InitSend( )** to send an Nb UP initialization message to the remote endpoint. Construct the IPM_NBUP_INIT_SEND structure with a list of two IPM_NBUP_RFCI_INFO structures.

  The first IPM_NBUP_RFCI_INFO structure, which represents the first RFCI, should contain three IPM_NBUP_SUBFLOW_INFO structures. For the first IPM_NBUP_SUBFLOW_INFO structure, specify NBUP_FLOW_SIZE_81_BITS. For the second structure, specify NBUP_FLOW_SIZE_103_BITS. For the third structure, specify NBUP_FLOW_SIZE_60_BITS. This represents the RFCI used for the source rate of AMR-NB 12.2 Kbps.

  The second IPM_NBUP_RFCI_INFO structure, which represents the second RFCI, should contain three IPM_NBUP_SUBFLOW_INFO structures. For the first IPM_NBUP_SUBFLOW_INFO structure, specify NBUP_FLOW_SIZE_39_BITS. For the second and third structures, specify NBUP_FLOW_SIZE_0_BITS. This represents the RFCI used for the source rate of AMR-NB SID.

  The IPMEV_INIT_SEND termination event indicates successful completion of the **ipm_InitSend( )** function.
- Upon receiving the IPMEV_INIT_RECEIVED event, the remote endpoint calls **ipm_InitResponseSend( )** to respond to the Nb UP message sent by the local endpoint.

The IPMEV_INIT_RESPONSE_SEND termination event indicates successful completion of the **ipm_InitResponseSend( )** function.

- Both the remote endpoint and the local endpoint should receive the IPMEV_INIT_COMPLETE unsolicited event, which indicates that the Nb UP session is successfully established. Upon receiving this event, the endpoints may exchange data, for example, using **mm_Play( )** and **mm_Record( )**.

1.9.3 Guidelines for Streaming G.711 (5 ms) over Nb UP

To stream G.711 (5 ms) audio over Nb UP, follow these general guidelines:

- Specify a native connection between the IPM device and the MM device. Native play and record operation takes place using native G.711 audio files.

  Note: For play operation, native audio files containing G.711 20 ms format may be used. The Dialogic® product transforms this data to 5 ms format before transmitting to the IP network.

- Where available, use the inline function to initialize a data structure. For example, use INIT_IPM_NBUP_PROFILE_INFO to initialize IPM_NBUP_PROFILE_INFO, and INIT_IPM_NBUP_INIT_SEND to initialize IPM_NBUP_INIT_SEND.
- Call **ipm_StartMedia( )** to start the session. In the IPM_NBUP_PROFILE_INFO structure, specify NBUP_PROFILE_G711ALAW64K_5MS or NBUP_PROFILE_G711ULAW64K_5MS in the eProfileType field.
- Call **ipm_InitSend( )** to send an Nb UP initialization message to the remote party. Construct the IPM_NBUP_INIT_SEND structure with a list of one IPM_NBUP_RFCI_INFO structure.

  The IPM_NBUP_RFCI_INFO structure should contain three IPM_NBUP_SUBFLOW_INFO structures. For the first IPM_NBUP_SUBFLOW_INFO structure, specify NBUP_FLOW_SIZE_320_BITS. For the second and third structures, specify NBUP_FLOW_SIZE_0_BITS. This represents the RFCI used for the source rate of G.711 5 ms.

  The IPMEV_INIT_SEND termination event indicates successful completion of the **ipm_InitSend( )** function.

- Upon receiving the IPMEV_INIT_RECEIVED event, the remote endpoint calls **ipm_InitResponseSend( )** to respond to the Nb UP message sent by the local endpoint.

  The IPMEV_INIT_RESPONSE_SEND termination event indicates successful completion of the **ipm_InitResponseSend( )** function.

- Both the remote endpoint and the local endpoint should receive the IPMEV_INIT_COMPLETE unsolicited event, which indicates that the Nb UP session is successfully established. Upon receiving this event, the endpoints may exchange data, for example, using **mm_Play( )** and **mm_Record( )**.

1.9.4 Guidelines for Streaming G.711 (20 ms) over Nb UP

To stream G.711 (20 ms) audio over Nb UP, follow these general guidelines:

- Specify a native connection between the IPM device and the MM device. Native play and record operation takes place using native G.711 audio files.
- Where available, use the inline function to initialize a data structure. For example, use INIT_IPM_NBUP_PROFILE_INFO to initialize IPM_NBUP_PROFILE_INFO, and INIT_IPM_NBUP_INIT_SEND to initialize IPM_NBUP_INIT_SEND.
- Call **ipm_StartMedia( )** to start the session. In the IPM_NBUP_PROFILE_INFO structure, specify NBUP_PROFILE_G711ALAW64K_20MS or NBUP_PROFILE_G711ULAW64K_20MS in the eProfileType field.
- Call **ipm_InitSend( )** to send an Nb UP initialization message to the remote party. Construct the IPM_NBUP_INIT_SEND structure with a list of one IPM_NBUP_RFCI_INFO structure.

  The IPM_NBUP_RFCI_INFO structure should contain three IPM_NBUP_SUBFLOW_INFO structures. For the first IPM_NBUP_SUBFLOW_INFO structure, specify NBUP_FLOW_SIZE_1280_BITS. For the second and third structures, specify NBUP_FLOW_SIZE_0_BITS. This represents the RFCI used for the source rate of G.711 20 ms.

  The IPMEV_INIT_SEND termination event indicates successful completion of the **ipm_InitSend( )** function.
- Upon receiving the IPMEV_INIT_RECEIVED event, the remote endpoint calls **ipm_InitResponseSend( )** to respond to the Nb UP message sent by the local endpoint.

  The IPMEV_INIT_RESPONSE_SEND termination event indicates successful completion of the **ipm_InitResponseSend( )** function.
- Both the remote endpoint and the local endpoint should receive the IPMEV_INIT_COMPLETE unsolicited event, which indicates that the Nb UP session is successfully established. Upon receiving this event, the endpoints may exchange data, for example, using **mm_Play( )** and **mm_Record( )**.

1.9.5 API Library Changes

In the Dialogic® IP Media Library API, the following values are added to the eProfileType field, eIPM_NBUP_PROFILE_TYPE enumeration, for the IPM_NBUP_PROFILE_INFO structure:

NBUP_PROFILE_AMR_NB
AMR-NB type connection

NBUP_PROFILE_G711ALAW64K_5MS
G.711 A-law 5 ms type connection

NBUP_PROFILE_G711ALAW64K_20MS
G.711 A-law 20 ms type connection

NBUP_PROFILE_G711ULAW64K_5MS
G.711 mu-law 5 ms type connection

NBUP_PROFILE_G711ULAW64K_20MS
G.711 mu-law 20 ms type connection

In the Dialogic® IP Media Library API, the following values are added to the eFlowSize field, eIPM_NBUP_FLOW_SIZE enumeration, for the IPM_NBUP_SUBFLOW_INFO structure:

NBUP_FLOW_SIZE_39_BITS
use 39 bits (for AMR-NB)

NBUP_FLOW_SIZE_60_BITS
use 60 bits (for AMR-NB)

NBUP_FLOW_SIZE_81_BITS
use 81 bits (for AMR-NB)

NBUP_FLOW_SIZE_103_BITS
use 103 bits (for AMR-NB)

NBUP_FLOW_SIZE_1280_BITS
use 1280 bits (for G.711)

1.9.6 Documentation

The online bookshelf contains information about all release features including features for application development, configuration, administration, and diagnostics.

For more information about the Dialogic® IP Media Library API, see the following documents:

- *Dialogic® IP Media Library API Programming Guide*
- *Dialogic® IP Media Library API Library Reference*

1.10 H.263 Using RFC 2429 (RFC 4629) Packetization

Service Update 107 adds support for streaming H.263 Baseline Profile (Profile 0) Level 10 video using RFC 2429 packetization from an RTSP (Real-time Streaming Protocol) Server to 3G-324M calls. Note that this codec is also known as H.263-1998 or H.263+. Only Baseline Profile is supported. No H.263 annexes are supported.

Previously only H.263 video using RFC 2190 packetization was supported.

1.10.1 Feature Description

This feature allows an application to natively stream audio and video from an RTSP Server to 3G-324M calls terminated by the Dialogic® product. These 3G-324M calls may be established over the PSTN (for example, E1 circuit-switched connection) or over IP.

For 3G-324M calls established over the PSTN, this feature uses DTI and M3G devices. For 3G-324M calls established over IP, this feature uses IPM and M3G devices. Note that the DTI device and the IPM device establish the transport to the remote endpoint. H.223 multiplexed multimedia data flows between these devices and the M3G device. The M3G device performs the multiplex/demultiplex operations and acts as the portal to the rest of the multimedia server.

The audio streaming formats supported for this feature include AMR-NB and G.723.

1.10.2 Usage Guidelines

The following usage guidelines as well as restrictions and limitations are described for this feature:

- This feature is only supported for the 3PCC/SIP call model.
- Although the video format streamed from the RTSP Server to the product is H.263-1998 using RFC 2429 (RFC 4629) packetization, only Baseline H.263 is supported. Annexes are not supported. Video transcoding is not supported. Specify **native connection** (no transcoding) between devices in the DM_PORT_CONNECT_INFO structure of the Device Management API library.
- Audio transcoding is not supported. Therefore, the audio format selected for the 3G-324M call must be the same as the audio format selected for the RTSP Server. Specify **native connection** (no transcoding) between devices in the DM_PORT_CONNECT_INFO structure of the Device Management API library.
- The RTSP Server must be configured appropriately to interoperate with this feature. The characteristics of the video streamed from the RTSP Server must be compatible with 3G mobile devices and the transport; namely, set to QCIF picture format and constant bit rate (CBR) mode rather than variable bit rate (VBR) mode. Bit rates should not exceed 40 Kbps and frame rates should not exceed 15 fps. A nominal bit rate and frame rate to use is 37.8 Kbps at 7.5 fps.
- This feature only supports half-duplex streaming from the IPM device to the 3G-324M device.
- This feature does not support video play and record operations for IPM calls.
- When switching between video sources for transmission of video to the 3G remote terminal, such as switching between the RTSP Server and the MM device, you must send an I-frame as the first video frame streamed from the new source. Ensure that this takes place by starting a new play from the MM device when it becomes the new source or by starting a new play from the RTSP Server when it becomes the new source.

1.10.3 API Library Changes

In the Dialogic® IP Media Library API, the following value is added to the eCoderType field, eIPM_CODER_TYPE enumeration, for the IPM_VIDEO_CODER_INFO structure:

CODER_TYPE_H263_1998
H.263-1998 (also known as H.263+) video coder

In the Dialogic® 3G-324M API, the M3G device is enhanced to receive Baseline H.263-1998 video data. This change is transparent to the user.

1.10.4 Documentation

The online bookshelf contains information about all release features including features for application development, configuration, administration, and diagnostics.

For more information about the Dialogic® IP Media Library API, see the following documents:

- *Dialogic® IP Media Library API Programming Guide*
- *Dialogic® IP Media Library API Library Reference*

Release Issues 2

The table below lists issues that can affect the hardware and software supported in the Dialogic® Multimedia Kit Software Release 1.0 for PCIe. The following information is provided for each issue:

Issue Type

This classifies the type of release issue based on its effect on users and its disposition:

- Known – A minor hardware or software issue. This category includes interoperability issues and compatibility issues. Known issues are still open but may or may not be fixed in the future.
- Known (permanent) – A known hardware or software issue or limitation that is not intended to be fixed in the future.
- Resolved – A hardware or software issue that was resolved (usually either fixed or documented) in this release.

Defect No.

A unique identification number that is used to track each issue reported via a formal Change Control System. Additional information on defects may be available via the Defect Query tool at *http://membersresource.dialogic.com/defects/*.
Note that when you select this link, you will be asked to either LOGIN or JOIN.

SU No.

For defects that were resolved in a Service Update, the Service Update number is shown. For defects that were resolved when the base release was generally available (before any Service Updates), a "--" is shown. For non-resolved issues, this information is left blank.

Product or Component

The product or component to which the problem relates; for example, an API.

Description

A summary description of the issue. For non-resolved issues, a workaround is included when available.

Issues Sorted by Type, Dialogic® Multimedia Kit Software Release 1.0 for PCIe

Issue Type	Defect No.	SU No.	Product or Component	Description
Resolved	IPY00078621	107	SNMP	When running query ipmsLicenseFileSelected and ipmsLicenseExpirationDate continuously, the system results in snmpwalk hang.
Resolved	IPY00045522	107	SNMP	Some MIB files with underscore characters may cause warnings and issues when used with netsnmptools.
Resolved	IPY00045305	107	Conferencing	Excessive CNF audio conferencing failures may occur when running at high densities.

Issues Sorted by Type, Dialogic® Multimedia Kit Software Release 1.0 for PCIe (Continued)

Issue Type	Defect No.	SU No.	Product or Component	Description
Resolved	IPY00044834	107	3G-324M	When M3G trace with logfile is enabled, this may cause a system failure if the logfile reaches over 2GB.
Resolved	IPY00044824	107	3G-324M	When WNSRP is enabled, **m3g_OpenLC( )** returns M3GEV_OPEN_LC_FAIL event with M3G_E_ERR_PROTOCOL error for Motorola handsets.
Resolved	IPY00044656	107	Multimedia	System may fail if application negotiates video size as QCIF and then incorrectly plays a CIF file.
Resolved	IPY00044655	107	Multimedia	With certain handsets, transcoding from H.263 to MPEG-4 or from MPEG-4 to MPEG-4, the audio quality may have crackling.
Resolved	IPY00044505	107	3G-324M	Using G.723.1 with certain 3G handsets may cause a system failure.
Resolved	IPY00043272	107	Firmware	When video transcoding mode is used, the I-Frame update requests are not processed properly. This would result in some Nokia phones generating excessive I-Frame update requests and ultimately causing video quality degradation.
Known (permanent)	IPY00045179		3G-324M	Momentary freeze or video degradation may occur when natively streaming files with video frames larger than 1000 bytes. ***Workaround:*** Refer to the Video Quality Considerations section in the *3G-324M API Programming Guide and Library Reference* or enable video transcoding.
Known (permanent)	IPY00043792		CLI	OA&M interfaces (CLI/SNMP) will not prevent a user from stopping IpMedia services even if an application is running and using IpMedia resources.
Known (permanent)	IPY00043739		Multimedia	When running **mm_Record( )** and **mm_Play( )**, it can't be issued at the same time without waiting for the ACK event.
Known (permanent)	IPY00042631		Firmware	Some audio degradation may be experienced using the QCELP audio codec at high densities.
Known (permanent)	IPY00042620		Voice	Performing transaction record results in missing random digits.
Known (permanent)	IPY00042570		Voice	SV_ABSPOS is not supported in the voice library. The host library throws an exception and only allows absolute adjustments set to 0 which is incorrect.
Known (permanent)	IPY00042556		Conferencing	The conferencing party attribute “DTMF clamping” is overwritten by conference level attribute when the party is removed and added back into the same conference.

Issues Sorted by Type, Dialogic® Multimedia Kit Software Release 1.0 for PCIe (Continued)

Issue Type	Defect No.	SU No.	Product or Component	Description
Known (permanent)	IPY00042520		DTMF	RFC2833 cannot be dynamically changed due to the DSP configuration. Any other RFC2833 payload type besides 101 will result in no digits received at all. RFC2833 only supports payload type 101. ***Workaround:*** The payload type must be set prior to start media.
Known	IPY00078715		Multimedia	**ipm_StopMedia( )** may not receive the complete event if **mm_Record( )** or **mm_Play( )** is set to run more than once in the same IPM session. ***Workaround:*** The following parameters must be called prior to an **ipm_StartMedia( )**: PARMCH_DTMFXFERMODE, PARMCH_RFC2833EVT_RX_PLT, PARMCH_RFC2833EVT_TX_PLT
Known	IPY00078650		3G-324M	Empty TCS is not supported in Dialogic® Multimedia Kit Software Release 1.0 for PCIe.
Known	IPY00078401		Voice	Occasional poor audio quality may occur with G.726 codecs.
Known	IPY00044062		Firmware	One may see the following error message logged during download. `ERROR - unexpected message, code 0x3, data=0x800082` ***Workaround:*** None required. Ignore message.
Known	IPY00043996		Firmware	The following message may be logged during execution. `QERROR_WARNING(2) cfc1.c e44 qkernerr.h 0` ***Workaround:*** None required. Ignore message.
Known	IPY00043982		RTP	While dialing from HMP hairpinned through MMK with G.729AB codecs, some DTMF digits may be missed.
Known	IPY00043725		Global Call	If **gc_TransmitAlarms( )** fails due to an empty alarm list, the application may not receive any notification of the failure.
Known	IPY00043528		Firmware	One may see the following message logged during execution. `could not write packet to JB` ***Workaround:*** None required. Ignore message.
Known	IPY00043475		Firmware	One may occasionally see the following message logged. `PSTREAM_Read ERROR` ***Workaround:*** None required. Ignore message.
Known	IPY00043470		Fax	There may be underrun error messages when using fax with a DNI board.

Issues Sorted by Type, Dialogic® Multimedia Kit Software Release 1.0 for PCIe (Continued)

Issue Type	Defect No.	SU No.	Product or Component	Description
Known	IPY00042811		Configuration	When trying to set Transparent Mode on a channel, **gc_SetConfigData( )** should return the proper events but ends up failing.
Known	IPY00042614		Fax	Data may not be received during a T.30 fax session when the remote side has a disconnected error.

Documentation Updates 3

The documentation updates for Dialogic® Multimedia Kit Software Release 1.0 for PCIe are divided into the following sections, which correspond to the top level categories used on the online documentation navigation page:

3.1 System Release Documentation

This section contains updates to the following documents (click the title to jump to the corresponding section):

- Dialogic® Multimedia Kit Software Release 1.0 for PCIe Release Guide

3.1.1 Dialogic® Multimedia Kit Software Release 1.0 for PCIe Release Guide

There are currently no updates to this document.

Note: A new version of the document (05-2635-002) is now available on the documentation bookshelf. See the Revision History section of the document for a description of the changes.

3.2 Installation and Configuration Documentation

This section contains updates to the following documents (click the title to jump to the corresponding section):

- Dialogic® Multimedia Kit Software Release 1.0 for PCIe Software Installation Guide
- Dialogic® System Configuration Guide
- Dialogic® Global Call CDP Configuration Guide

3.2.1 Dialogic® Multimedia Kit Software Release 1.0 for PCIe Software Installation Guide

There are currently no updates to this document.

3.2.2 Dialogic® System Configuration Guide

There are currently no updates to this document.

3.2.3 Dialogic® Global Call CDP Configuration Guide

There are currently no updates to this document.

3.3 OA&M Documentation

This section contains updates to the following documents (click the title to jump to the corresponding section):

- Dialogic® Host Media Processing Diagnostics Guide

3.3.1 Dialogic® Host Media Processing Diagnostics Guide

Update to Diagnostics Guide.

A new chapter titled Chapter 6, Telecom Subsystem Summary Tool (its_sysinfo) Reference should be added with the following content:

Chapter 6, Telecom Subsystem Summary Tool (its_sysinfo) Reference

This chapter describes the Telecom Subsystem Summary Tool (its_sysinfo) and the information it collects. This chapter contains the following information about its_sysinfo:

Description

The Telecom Subsystem Summary Tool (its_sysinfo) provides a simple way to collect information about systems built using Dialogic® products. The its_sysinfo tool collects data from the system on which you execute it and provides you with information about the system environment: the operating system, computer architecture, Dialogic® HMP software, and operational logs.

The its_sysinfo tool also enables you to collect baseline information about the system for quick review of configuration issues when determining system configuration consistency. This information is collected in a file, compressed, and archived as part of the complete system information collection in an archive file named *its_sysinfo.zip* (or a name you specify). If the installed system is configured in such a way that the baseline information is not available, the its_sysinfo tool will indicate “No Information Available.”

In addition to the standard information captured by the its_sysinfo tool, you can manually add files to the archive that is created by *its_sysinfo* after you run the tool so you can preserve additional information that might help with resolving issues.

Command Line Interface

On the command line, enter `its_sysinfo filename` where *filename* is the name you want to give the zip file (it can be the complete path). The its_sysinfo tool will collect system information and compress it into the zip file. If you do not specify any filename, then the information gets compressed in a zip file with the default name *its_sysinfo.zip*.

Information Collected by its_sysinfo

The following information is collected under *its_sysinfo.htm*, which is one of the files that is added to the archive:

- **General System Information**
  - **Environment Variables** – information about the operating system environment variables
  - **System Event Logs** – See **Table 1**.
  - **Memory and Processor** – information about the platform's available and used memory and CPU type and number as of the time you executed the its_sysinfo tool
  - **Operating System** – information about the operating system's version, service pack, and language
  - **/proc/meminfo file** – Linux only
- **Information Specific to Dialogic® Products**
  - **Linux Package Info** – For Linux systems, its_sysinfo provides a Linux Package Info section that shows installed RPM information and selections made from the install menu.
  - **Windows® Package Info** – For Windows® systems, its_sysinfo provides a Windows Package Info section that shows the following information for the active Dialogic® HMP Software and for the previously installed Dialogic® HMP Software: release name and build number, build type, install location, install date, installed by, and installed features.
  - **Installed Devices** – its_sysinfo collects information about devices detected in Dialogic® system configurations and startup.
  - **Configuration Settings for Installed Devices** – its_sysinfo captures the stored values used by the configuration tool for Dialogic® system configurations and startup. For more information about the configuration tool, refer to the configuration guide(s) for the Dialogic® HMP Software.
  - **Firmware Files and Versions** – its_sysinfo collects information about the files listing all firmware file names and version numbers.
  - **FCD Files** – its_sysinfo collects information about the*.fcd* files used by the configuration tool for Dialogic® system configurations and startup. For more information about FCD files, refer to the configuration guide(s) for the Dialogic® HMP Software.
  - **PCD Files** – its_sysinfo collects information about the*.pcd* files used by the configuration tool for Dialogic® system configurations and startup. For more information about PCD files, refer to the configuration guide(s) for the Dialogic® HMP Software.

  - **CONFIG Files** – its_sysinfo collects information about the.*config* files used by the configuration tool for Dialogic® system configurations and startup. For more information about CONFIG files, refer to the configuration guide(s) for the Dialogic® HMP Software.
  - **Global Call Configuration** – its_sysinfo collects information about the Global Call PDK subsystem configuration, which is contained in the *pdk.cfg* file. This file specifies the Global Call protocol modules and the country dependent parameter settings downloaded to each device. For more information about Global Call configuration, refer to the *Dialogic® Global Call CDP Configuration Guide*.
  - **Build Information** – its_sysinfo collects information about the contents of the *buildinfo.ini* file used for the installed Dialogic® system. The *buildinfo.ini* file contains information about what is in the build of the software.
  - **Board Memory Dumps**
  - **Event Viewer Data** – its_sysinfo collects information about the last events reported to the operating system and to the event logger from the Dialogic® system.
  - **File Versions**
- **Extra data collected with the troubleshoot script includes the following:**
  - Network card configuration and current network settings such as routes and arp cache.
  - PCI card settings.
  - CLI information such as license and interface.

The its_sysinfo tool also checks for the log files listed in Table 1 and adds them to the archive.

Note: It is possible to specify a name for some log files. However, its_sysinfo only collects files with default names.

Table 1. Log Files Archived by its_sysinfo

Log Files	Windows®	Linux
OA&M files	$(SystemRoot)\System32\anm_debug.log $(SystemRoot)\System32\anm_trace.log %INTEL_DIALOGIC_DIR%\log\ClusterPkg.log %INTEL_DIALOGIC_DIR%\log\ClusterPkg.log.0 *or* $(SystemRoot)\System32\ClusterPkg.log* $(SystemRoot)\System32\confslot.log %INTEL_DIALOGIC_DIR%\log\ctbb*.log *or* $(SystemRoot)\System32\ctbb*.log dlgcInstall.log is in $(TEMP) %INTEL_DIALOGIC_DIR%\log\dlgcsyslogger.log %INTEL_DIALOGIC_DIR%\log\DM3AutoDump.log $(SystemRoot)\System32\dm3bsp.log $(SystemRoot)\System32\dm3fdspdll.log $(SystemRoot)\System32\frustatus.log %INTEL_DIALOGIC_DIR%\log\GenLoad.log %INTEL_DIALOGIC_DIR%\log\merc.log *or* $(SystemRoot)\System32\merc.log %INTEL_DIALOGIC_DIR%\log\ncm.ini %INTEL_DIALOGIC_DIR%\log\oam.log %INTEL_DIALOGIC_DIR%\log\Sctsassi.log	$(INTEL_DIALOGIC_DIR)/log/board*.log $(INTEL_DIALOGIC_DIR)/log/clusterpkg.log $(INTEL_DIALOGIC_DIR)/log/clusterpkg.log.* $(INTEL_DIALOGIC_DIR)/log/dlgsyslogger.log $(INTEL_DIALOGIC_DIR)/log/genload.log $(INTEL_DIALOGIC_DIR)/log/iptconf.log $(INTEL_DIALOGIC_DIR)/log/oam.log

Table 1. Log Files Archived by its_sysinfo (Continued)

Log Files	Windows®	Linux
Demos	%INTEL_DIALOGIC_DIR%\log\rgademo.log %INTEL_DIALOGIC_DIR%\log\Board[#].log	$(INTEL_DIALOGIC_DIR)/log/rgademo.log $(INTEL_DIALOGIC_DIR)/log/Board[#].log $(INTEL_DIALOGIC_DIR)/log/dlgrhdemo.log
RTF log files	<Logfile path>/rtflog*.txt <Logfile path> found in $(INTEL_DIALOGIC_DIR)\cfg\RtfConfigwin.xml or $(DLCFGPATH)\rtfconfig.xml	<Logfile path>/rtflog*.txt <Logfile path> specified in $(INTEL_DIALOGIC_DIR)/cfg/RtfConfigLinux.xml
	<Logfile path>/rtflog.txt <Logfile path> specified in $(DLCFGPATH)\rtfconfig.xml	<Logfile path>/rtflog*.txt <Logfile path> specified in $(DLGCFGPATH)/RtfConfigLinux.xml
CASTrace log files	$(INTEL_DIALOGIC_DIR)\log\CAStrace.log	$(INTEL_DIALOGIC_DIR)/log/CAStrace.log.*
Debug Angel files	$(INTEL_DIALOGIC_DIR)\bin\DebugAngel.log	$(INTEL_DIALOGIC_DIR)/log/debugangel.*
Pstndiag Trace log files	$(INTEL_DIALOGIC_DIR)\log\pstndiag.*	$(INTEL_DIALOGIC_DIR)/log/pstndiag.*
IP Protocol files	$(SystemRoot)\System32\gc_h3r.log *or* $(INTEL_DIALOGIC_DIR)\bin\gc_h3r.log $(SystemRoot)\System32\rtvsp1.log *or* $(INTEL_DIALOGIC_DIR)\bin\rtvsp1.log $(SystemRoot)\System32\sdplog.txt *or* $(INTEL_DIALOGIC_DIR)\bin\sdplog.txt $(SystemRoot)\System32\siplog.txt *or* $(INTEL_DIALOGIC_DIR)\bin\siplog.txt **Note:** This is obsolete. IP protocols support RTF.	$(HOME)/g*.log $(HOME)/rtvsp1.log **Note:** This is obsolete. IP protocols support RTF.
Dr. Watson dump and log files	$(USERPROFILE)\Local Settings\Application Data\Microsoft\Dr Watson\DrWtsn32.log	
Its_sysinfo logfile files	its_sysinfo.log	its_sysinfo.log

Notes: ***1.*** Windows®: INTEL_DIALOGIC_DIR=C:\Program Files\Dialogic and SystemRoot=C:\WINNT

2. Linux: INTEL_DIALOGIC_DIR=/usr/dialogic/ and HOME=/root

System Information Data Structuring

Data collected by its_sysinfo will be available as a single compressed file that contains the following:

- A master data file in HTML format.
- 0 or more files as attachments. The HTML file will contain links and explanations of the attachments.
- An application operation log file generated the during the data collecting process.
- The operation log files and attachments will be put in the compressed zip file.

Update to Diagnostics Guide.

A new chapter titled Chapter 7, DM3post Reference should be added with the following content:

Chapter 7, DM3post Reference

This chapter provides reference information about the DM3post tool. The following topics are included:

Description

The DM3post tool, sometimes referred to as “POST-on-demand”, can perform diagnostics on a stopped board at any time to detect and isolate possible hardware faults.

Note: This tool can be run on Dialogic® DNI boards.

Guidelines

The board must be in a “stopped” state before the DM3post tool can be run. If you do not use the reset option (`-r`), DM3post will *not* reset the board and will only retrieve the POST results for the last reset that occurred. If you use DM3post with the reset option, DM3post will force a full reset of the specified board, forcing the Control Processor (CP) POST diagnostics to run. DM3post will then retrieve the POST results from the SRAM and provide a PASS/FAIL indication to you. The board will remain in a stopped state, so you must restart the board.

Note: For the slot number, provide the physical slot number. This information may be obtained from the listboards utility, which is described in the *Dialogic® Host Media Processing Diagnostics Guide* for the release.

Options

The following command line options are used with the DM3post tool:

-s<n>

slot number (required). Use the listboards utility to obtain the board’s physical slot number.

Note: The listboards utility is described in the *Dialogic® Host Media Processing Diagnostics Guide* for the release.

-b<n>
: bus number (optional if there is only one bus *or* if the slot number is unique)

-l
: logs event (optional). Output is logged to the screen.

-q
: suppresses output (optional). The tool operates in silent mode.

-r
: resets board before retrieving diagnostics results (optional). If not set, results displayed will be those generated at board startup.

-h
: displays the tool's help screen

-v
: displays the program version

Example 1: Run DM3post on a Board in Slot 17, Bus 0

The following example runs DM3post on a board in slot 17, bus 0:

```
dm3post -s17 -b0
```

You will get the following response:

```
You have chosen to read the initial POST diagnostic results from the board in slot 17, bus 0. No
board reset will occur.

Do you wish to continue (y/n)?
```

If you answer Y, the following will be printed to the screen:

```
Retrieving results...
```

The success/failure message will be printed to the screen when POST is complete. Here is an example:

```
SUCCESS: POST passed for board in slot 17, bus 0. Diagnostic Codes: 0x03 0xfc
```

Example 2: Run DM3post with the Reset Option on a Board in Slot 17, Bus 0

The following example runs DM3post with the reset option on a board in slot 17, bus 0:

```
dm3post -s17 -b0 -r
```

You will get the following response:

```
You have chosen to run diagnostics on the board in slot 17, bus 0.

Do you wish to continue (y/n)?
```

If you answer Y, the following will be printed to the screen:

```
dm3post processing...
```

The success/failure message will be printed to the screen when POST is complete. Here is an example:

```
SUCCESS: POST passed for board in slot 17, bus 0. Diagnostic Codes: 0x03 0xfc
```

Update to Diagnostics Guide.

A new chapter titled Chapter 8, List Board Information Utility (listboards) should be added with the following content:

Chapter 8, List Board Information Utility (listboards)

Note: You must install the DLGCdmdev RPM to use the List Board Information utility regardless of the type of board in your system. To install the DLGCdmdev RPM, select a menu item during installation of the software release (see the *Software Installation Guide*).

The List Board Information utility displays information about the Dialogic® telecom boards that are installed in the system.

The List Board Information utility has two levels: Level 1 and Level 2 (referred to as L1 and L2 for convenience). Each level has a separate parameter list. See Table 2 for a list of L1 parameters and Table 3 for a list of L2 parameters.

Note: The List Board Information utility Level 1 supports Dialogic® DM3 architecture boards only. The List Board Information utility Level 2 supports Dialogic® DM3 architecture boards and Springware architecture boards.

L1 is invoked using the `-l1` or `--release 1` parameter. L1 is also invoked by default (when you invoke the utility without any parameters or without the L2 parameter).

Command Line: `listboards [-l1 | --release1] [level1_parameter_list]`

L2 is invoked using the `-l2` or `--release 2` parameter.

Command Line: `listboards -l2  | --release2 [level2_parameter_list]`

The following tips and hints on using the List Board Information utility are provided:

- This utility accepts parameters in two synonymous forms: abbreviated (`-x`) and verbose (`--yyyyy`). Examples: `-h` or `--help` to display help; `-v` or `--version` to find out the utility version; `-r` or `--oROM` for ROM information; `-l` or `--release` to specify the release level. Some parameters are available in both forms, but others are available only in one form; for example: `--oLID` and `--oPHYS` have no abbreviated form.
- Parameters must always be preceded by the dash (-) or double dash (--); otherwise the text will be ignored.
- Verbose parameters require that a space be used between the parameter and its specified value; for example, `listboards -l2 --iAUID 50023`. Abbreviated parameters can be used with or without the space.
- Options that are described as "intrusive" send messages to the board and thus may temporarily affect performance.

- This utility displays or provides a logical ID (Log ID) number, which is synonymous with board number and is specified when running any of the administrative utilities that require a logical ID number. The List Board Information utility displays the logical board number along with the serial number of the board. Board numbers are dynamically assigned on start up, so use the serial number to physically identify the board in the chassis. Also, the `--oLID` parameter is used to obtain the logical ID number of a specified board.
- This utility displays a list of all boards if no parameters are used to identify a specific board. For example, `listboards -l 2` displays a list of all boards, while `listboards -l 2 -a 50023 --oLID` returns the logical board ID for the board having AUID 50023.
- More complete information is displayed by the List Board Information utility if you start the boards first.

Table 4 provides a list of L1 parameters.

Table 2. Listboards Level 1 Parameters

Parameter	Description
-b <*boardnumber*> --scbus <*boardnumber*>	Lists SCBus information for board with the specified logical ID (intrusive). **Note:** The -b option performs a different function in L2.
-c <*pcislotnumber*> --cfgrom <*pcislotnumber*>	Lists ROM configuration from specified PCI slot number.
-h --help	Displays help. Performs the same function in L1 and L2.
-i <*boardnumber*> --info <*boardnumber*>	Retrieves detailed hardware information for board with the specified logical ID (intrusive).
-l <*level*> --release <*level*>	Selects *listboards* release level to run: 1 (default) or 2. This table only describes L1 parameters. L1 supports Dialogic® DM3 architecture boards only.
-v --version	Lists utility version information. Performs the same function in L1 and L2.

Example: The following command runs the List Board Information utility using level 1:

```
listboards
```

See Figure 1 below for an example of the level 1 output.

Figure 1. Listboards Level 1 Example Output

```
BrdNum CfgId  Type Bus Slot PhysAddr   RamSize   Irq  State
   1     0     P    0    0  ff000000    80000    0xa  DOWNLOADED
   2     1     P    2    2  faf00000    80000    0xb  DOWNLOADED
   3     2     P    2    6  fae80000    80000    0xb  DOWNLOADED
```

Table 3 below provides a list of L2 parameters.

Table 3. Listboards Level 2 Parameters

Parameter	Description
-a <*auid*> --iAUID <*auid*>	Lists board with matching AUID. This parameter is also used with the --oLID or --oPHYS parameters.
-b <*pcibus*> -s <*pcislot*> --iPCIBus <*pcibus*> --iPCISlot <*pcislot*>	Lists board with matching PCI bus number and PCI slot number. Both parameters are required. These parameters can also be used with the --oLID or --oPHYS parameters. **Note:** The -b option performs a different function in L1, and the -s option is not supported in L1.
-e --extrainfo	Provides extra Dialogic® DM3 information on model number, driver configuration ID, driver state, and admin state. **Note:** Supported on Dialogic® DM3 architecture boards only.
-h --help	Displays help. Performs the same function in L1 and L2.
-l <*level*> --release <*level*>	Selects *listboards* release level to run: 1 (default) or 2. This table only describes L2 parameters. L2 supports Dialogic® DM3 architecture boards and Springware architecture boards
-p <*phys_id*> | <*wheel_id*> --iPHYS <*phys_id*> | <*wheel_id*>	Lists board with matching physical ID for Dialogic® CompactPCI or thumbwheel ID for PCI.
-q --quiet	Lists in quiet mode (without the title bar).
-r --oROM	Outputs ROM of found board. **Note:** Supported on Dialogic® DM3 architecture boards only.
-s <*pcislotnumber*> --iPCISlot <*pcislotnumber*>	Specifies the PCI slot number of a board (for --oLID or --oPHYS). This parameter must be used with the -b <*pcibus*> parameter. These parameters can also be used with the --oLID or --oPHYS parameters. **Note:** The -b option performs a different function in L1, and the -s option is not supported in L1.
-v --version	Lists utility version information. Performs the same function in L1 and L2.
<*board*> --oLID	Outputs logical ID of specified <*board*>. The board must be specified with the -a option or with the -b and -s options (otherwise it uses the first board by default). **Note:** This parameter must be used in the sequence shown; that is, the specified board (input) must precede the --oLID parameter (output).
<*board*> --oPHYS	Outputs physical ID (Dialogic® CompactPCI with Hot Swap Kit) or thumbwheel ID (PCI, or Dialogic® CompactPCI without Hot Swap Kit) of specified <*board*>. The board must be specified with the -a option or with the -b and -s options (otherwise it uses the first board by default). **Note:** This parameter must be used in the sequence shown; that is, the specified board (input) must precede the --oPHYS parameter (output).

Example: The following command runs the List Board Information utility using level 2:

```
listboards -l2
```

See Figure 2 below for an example of level 2 output.

Figure 2. Listboards Level 2 Example Output

```
| AUID |CT Platform|PCIBus|PCISlot|ThumbWheel|Log ID|Serial Num|ModelNum

 50002  DM3         0      13      0          1      KS014983   0x100
 50011  DM3         2      9       2          2      KS014980   0x100
 50020  DM3         2      10      6          3      GG000104   0x1E01
 50045  Springware  2      11      4          4      GP689090   <un>

Total number of boards: 4
```

Note: Display of the model number is not supported for Springware architecture boards.

3.4 Programming Libraries Documentation

This section contains updates to the following documents (click the title to jump to the corresponding section):

- Dialogic® 3G-324M API Programming Guide and Library Reference
- Dialogic® Conferencing API Library Reference
- Dialogic® Conferencing API Programming Guide
- Dialogic® Device Management API Library Reference
- Dialogic® Digital Network Interface Library Reference
- Dialogic® Fax Software Reference
- Dialogic® Global Call API Library Reference
- Dialogic® Global Call API Programming Guide
- Dialogic® Global Call E1/T1 CAS/R2 Technology Guide
- Dialogic® Global Call IP Technology Guide
- Dialogic® Global Call ISDN Technology Guide
- Dialogic® Global Call SS7 Technology Guide
- Dialogic® IP Media Library API Library Reference
- Dialogic® IP Media Library API Programming Guide
- Dialogic® Media Toolkit API Library Reference
- Dialogic® Multimedia API Programming Guide and Library Reference
- Dialogic® Standard Runtime Library API Library Reference
- Dialogic® Standard Runtime Library API Programming Guide
- Dialogic® Voice API Library Reference
- Dialogic® Voice API Programming Guide

3.4.1 Dialogic® 3G-324M API Programming Guide and Library Reference

There are currently no updates to this document.

Note: A new version of the document (05-2558-005) is now available on the documentation bookshelf. See the Revision History section of the document for a description of the changes.

3.4.2 Dialogic® Conferencing API Library Reference

There are currently no updates to this document.

3.4.3 Dialogic® Conferencing API Programming Guide

Update to Chapter 8, Building Applications.

Added the following sentence to Section 8.1, Compiling and Linking: When compiling an application, you must list Dialogic libraries before all other libraries such as operating system libraries.

3.4.4 Dialogic® Device Management API Library Reference

There are currently no updates to this document.

Note: A new document (05-2222-009) is now available on the documentation bookshelf. See the Revision History section of the document for a description of the changes.

3.4.5 Dialogic® Digital Network Interface Library Reference

There are currently no updates to this document.

3.4.6 Dialogic® Fax Software Reference

There are currently no updates to this document.

3.4.7 Dialogic® Global Call API Library Reference

There are currently no updates to this document.

3.4.8 Dialogic® Global Call API Programming Guide

Update to Chapter 12, Building Applications.

Added the following sentence to Section 12.1, Compiling and Linking: When compiling an application, you must list Dialogic libraries before all other libraries such as operating system libraries.

3.4.9 Dialogic® Global Call E1/T1 CAS/R2 Technology Guide

There are currently no updates to this document.

3.4.10 Dialogic® Global Call IP Technology Guide

There are currently no updates to this document.

3.4.11 Dialogic® Global Call ISDN Technology Guide

There are currently no updates to this document.

3.4.12 Dialogic® Global Call SS7 Technology Guide

There are currently no updates to this document.

Note: A new document (05-2274-005) is now available on the documentation bookshelf. See the Revision History section of the document for a description of the changes.

3.4.13 Dialogic® IP Media Library API Library Reference

IPM_NBUP_PROFILE_INFO data structure.

Connection types for AMR-NB and G.711 are added. For details, refer to AMR-NB and G.711 Audio Over Nb UP section in Post-Release Developments chapter.

IPM_NBUP_SUBFLOW_INFO data structure.

Flow sizes for AMR-NB and G.711 are added. For details, refer to AMR-NB and G.711 Audio Over Nb UP section in Post-Release Developments chapter.

IPM_VIDEO_CODER_INFO data structure.

The CODER_TYPE_H263_1998 is added for H.263 using RFC 2429 (RFC 4629) packetization. For details, refer to H.263 Using RFC 2429 (RFC 4629) Packetization section in Post-Release Developments chapter.

3.4.14 Dialogic® IP Media Library API Programming Guide

Using AMR coders for narrow band audio and using enhanced variable rate codecs (EVRC).

The following information should be added to the programming guide, possibly as a new chapter.

Using AMR Coders for Narrow Band Audio

Description

AMR is an adaptive multi-rate speech codec. During operation, both local and remote sides can request a change in the bit rate and dynamically adjust the bandwidth. The protocol uses the following:

- A Frame Type (FT) to indicate the transmitted bit rate
- A Codec Mode Request (CMR) value to request a particular bit rate in every packet

To control the bit rate, AMR assumes that all connections are bi-directional.

This feature is specific to AMR-NB and excludes support for AMR-WB and AMR-WB+, which support wideband audio and some other formats not addressed by AMR-NB.

The codec supports the following bit rates:

- 12.2 kbit/s (GSM EFR)
- 10.2 kbit/s
- 7.95 kbit/s
- 7.40 kbit/s (IS-641)
- 6.70 kbit/s (PDC-EFR)
- 5.90 kbit/s
- 5.15 kbit/s
- 4.75 kbit/s
- 1.80 kbit/s (assuming SID frames are continuously transmitted)

Note: The 1.80 kbit/s rate is not actually a voice signal, but the bit rate consumed when Voice Activation Detection (VAD) is processing a silence.

This feature is only supported when using the Session Initiated Protocol (SIP). None of the available SDP options are currently supported through Global Call (that is, direct first-party call control). The options are only available using third-party call control (3PCC), where the application is responsible for interpreting received SDP text strings and for constructing all outbound SDP text strings.

IP Media Library API Support

The bit rate can be controlled using a new media type in the IPM_MEDIA structure. The IP Media Library API allows the application to provide a preferred bit rate and a rule to determine how changes in the received CMR value control the transmitted bit rate.

Preferred bit rate

The preferred bit rate is specified by setting the eMediaType in the IPM_MEDIA structure to MEDIATYPE_AUDIO_REMOTE_CODER_INFO and setting the CoderInfo.eCoderType to the desired transmit bit rate.

CMR value

The transmitted CMR value is specified by setting the eMediaType in the IPM_MEDIA structure to MEDIATYPE_AUDIO_LOCAL_CODER_INFO and setting the CoderInfo.eCoderType to the desired CMR value.

The CMR rules are specified by a new media type called MEDIATYPE_AUDIO_REMOTE_CODER_OPTIONS_INFO. A new field AudioCoderOptionsInfo (of type IPM_AUDIO_CODER_OPTIONS_INFO) has been added to the IPM_MEDIA union. The CMR control is implemented by allowing the host application to specify one of two rules.

- The first rule, “CMR Tracking,” indicates that the transmit bit rate should follow the CMR value in the received packet.
- The second rule, “CMR Limit,” indicates that the transmit bit rate should follow the CMR value in the received packet with a maximum value specified by the preferred bit rate.

With both CMR rules, it is necessary to specify a preferred rate to avoid the specific case of determining what rate to transmit at before the first CMR value is received. The software will transmit at the preferred rate before the first packet is received or when a CMR value of 15 (don't care) is received from the opposite side.

The CMR rules are set in the AudioCoderOptionsInfo.unCoderOptions field by ORing in either CODER_OPT_AMR_CMR_TRACK or CODER_OPT_AMR_CMR_LIMIT.

Specifying a CMR rule is mandatory and the rules are mutually exclusive.

Note: The CMR rules are not used by the MEDIATYPE_AUDIO_LOCAL_CODER_OPTIONS_INFO eMediaTypes.

Supported RTP Payload Format

AMR supports two different formats for the RTP payload:

- “Bandwidth efficient,” to minimize the amount of network bandwidth.
- “Octet-aligned,” to make the packet parsing easier for the AMR application.

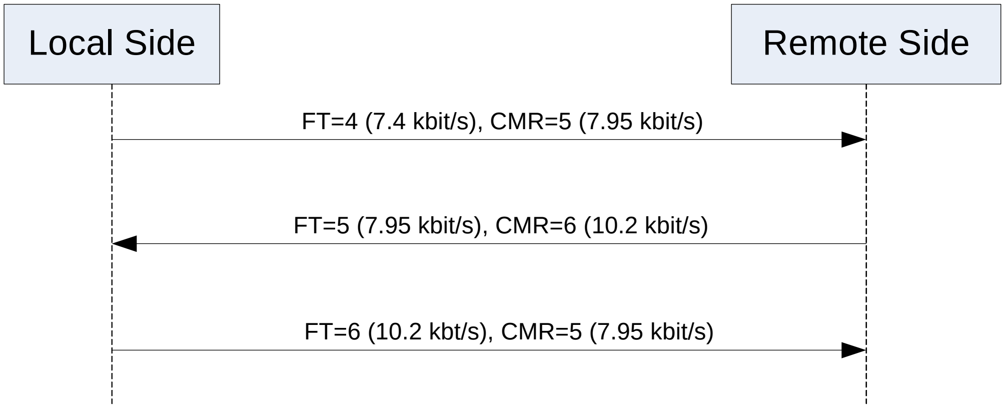
The AMR RTP payload format is specified in the new IPM_AUDIO_CODER_OPTIONS_INFO AudioCoderOptionsInfo structure (in the IPM_MEDIA structure). The MEDIATYPE_AUDIO_LOCAL_CODER_OPTIONS_INFO media type controls the receive side, and the MEDIATYPE_AUDIO_REMOTE_CODER_OPTIONS_INFO media type controls the transmit side.

The IPM_MEDIA structure contains a union with an IPM_AUDIO_CODER_OPTIONS_INFO structure. The AMR RTP payload format is controlled by ORing in either CODER_OPT_AMR_EFFICIENT or CODER_OPT_AMR_OCTET in the AudioCoderOptionsInfo.unCoderOptions field.

Note: Specifying an AMR RTP payload is mandatory and the formats are mutually exclusive.

Example 1

In this example, the host application wants to transmit at the bit rate requested by the incoming CMR value. The following diagram depicts this use case, where the local side is the host application.


Local Side
Remote Side
FT=4 (7.4 kbit/s), CMR=5 (7.95 kbit/s)
FT=5 (7.95 kbit/s), CMR=6 (10.2 kbit/s)
FT=6 (10.2 kbt/s), CMR=5 (7.95 kbit/s)

Note: While the diagram above implies an immediate reaction to a CMR from the other side, in reality, the other side’s response to a CMR may take a few packets.

The sequence of activities is as follows:

1. The host is transmitting at the preferred bit rate set by the host application via the remote audio coder settings, in this case 7.4 kbit/s.

2. The host sets its CMR value to 7.95 kbit/s via the local audio coder setting.

3. The host sets its RTP payload to efficient bandwidth and CMR tracking mode.

4. The host receives its first packet from the remote side with the CMR value higher than the preferred rate.

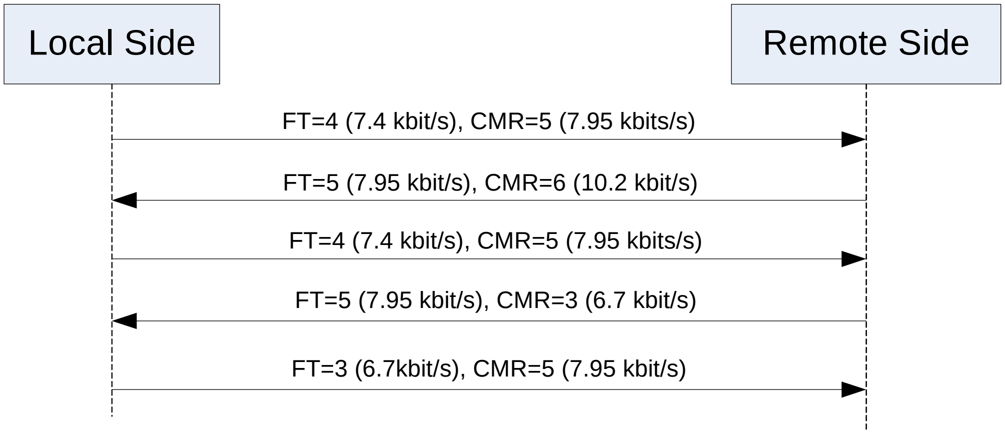
5. The host changes the transmitted bit rate to match the CMR value.

The following code demonstrates the configuration required to handle this scenario.

```
...
/* Setup IP address here */

// Local Audio Coder
ipmMediaInfo.MediaData[unCount].eMediaType = MEDIATYPE_AUDIO_LOCAL_CODER_INFO;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.eCoderType = CODER_TYPE_AMRNB_7_95k;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.eFrameSize = CODER_FRAMESIZE_20;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.unFramesPerPkt = 1;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.eVadEnable = CODER_VAD_ENABLE
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.unCoderPayloadType = 96;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.unRedPayloadType = 0
unCount++;

// Remote Audio Coder
ipmMediaInfo.MediaData[unCount].eMediaType = MEDIATYPE_AUDIO_REMOTE_CODER_INFO;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.eCoderType = CODER_TYPE_AMRNB_7_4k;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.eFrameSize = CODER_FRAMESIZE_20;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.unFramesPerPkt = 1;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.eVadEnable = CODER_VAD_ENABLE
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.unCoderPayloadType = 96;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.unRedPayloadType = 0
unCount++;

ipmMediaInfo.MediaData[unCount].eMediaType = MEDIATYPE_AUDIO_LOCAL_CODER_OPTIONS_INFO;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderOptionsInfo.unVersion =
                                                        IPM_AUDIO_CODER_OPTIONS_INFO_VERSION;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderOptionsInfo.unCoderOptions=
                                                        CODER_OPT_AMR_EFFICIENT;
unCount++;

ipmMediaInfo.MediaData[unCount].eMediaType = MEDIATYPE_AUDIO_REMOTE_CODER_OPTIONS_INFO;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderOptionsInfo.unVersion =
                                                        IPM_AUDIO_CODER_OPTIONS_INFO_VERSION;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderOptionsInfo.unCoderOptions=
                                               CODER_OPT_AMR_CMR_TRACK | CODER_OPT_AMR_EFFICIENT;
unCount++;
ipmMediaInfo.unCount = unCount;
...
```

Example 2

In this example, the host application wants to limit the transmitted bit rate to less than the preferred value. The following diagram shows this case, where the local side is the host application.


Local Side
Remote Side
FT=4 (7.4 kbit/s), CMR=5 (7.95 kbits/s)
FT=5 (7.95 kbit/s), CMR=6 (10.2 kbit/s)
FT=4 (7.4 kbit/s), CMR=5 (7.95 kbits/s)
FT=5 (7.95 kbit/s), CMR=3 (6.7 kbit/s)
FT=3 (6.7kbit/s), CMR=5 (7.95 kbit/s)

Note: While the diagram above implies an immediate reaction to a CMR from the other side, in reality, the other side's response to a CMR may take a few packets.

The sequence of activities is as follows:

1. The host is transmitting at the preferred bit rate set by the host application via the remote audio coder settings, in this case 7.4 kbit/s.

2. The host sets its initial CMR value to 7.95 kbit/s via the local audio coder setting.

3. The host sets its RTP payload to efficient bandwidth and CMR limit mode.

4. The host receives it first packet from the remote side with the CMR value higher than the preferred rate.

5. The host leaves the transmitted bit rate at the preferred value.

6. The host receives a packet from the remote side with a CMR value less than the preferred rate.

7. The host changes the transmitted bit rate to match the received CMR value.

The following code demonstrates the configuration required to handle this scenario.

```
...
/* Setup IP address here */

// Local Audio Coder
ipmMediaInfo.MediaData[unCount].eMediaType = MEDIATYPE_AUDIO_LOCAL_CODER_INFO;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.eCoderType =
CODER_TYPE_AMRNB_7_95k;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.eFrameSize = CODER_FRAMESIZE_20;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.unFramesPerPkt = 1;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.eVadEnable = CODER_VAD_ENABLE
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.unCoderPayloadType = 96;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.unRedPayloadType = 0
unCount++;

// Remote Audio Coder
ipmMediaInfo.MediaData[unCount].eMediaType = MEDIATYPE_AUDIO_REMOTE_CODER_INFO;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.eCoderType = CODER_TYPE_AMRNB_7_4k;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.eFrameSize = CODER_FRAMESIZE_20;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.unFramesPerPkt = 1;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.eVadEnable = CODER_VAD_ENABLE
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.unCoderPayloadType = 96;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderInfo.unRedPayloadType = 0
unCount++;

ipmMediaInfo.MediaData[unCount].eMediaType = MEDIATYPE_AUDIO_LOCAL_CODER_OPTIONS_INFO;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderOptionsInfo = {0};
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderOptionsInfo.unVersion =
                                                IPM_AUDIO_CODER_OPTIONS_INFO_VERSION;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderOptionsInfo.unCoderOptions=
                                              CODER_OPT_AMR_CMR_LIMIT | CODER_OPT_AMR_EFFICIENT;
unCount++;

ipmMediaInfo.MediaData[unCount].eMediaType = MEDIATYPE_AUDIO_REMOTE_CODER_OPTIONS_INFO;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderOptionsInfo = {0};
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderOptionsInfo.unVersion =
                                                IPM_AUDIO_CODER_OPTIONS_INFO_VERSION;
ipmMediaInfo.MediaData[unCount].mediaInfo.AudioCoderOptionsInfo.unCoderOptions=
                                                CODER_OPT_AMR_EFFICIENT;
unCount++
ipmMediaInfo.unCount = unCount;
```

Using Enhanced Variable Rate Codecs

Enhanced Variable Rate Codecs (EVRCs) as specified in *RFC 3558 - RTP Payload Format for Enhanced Variable Rate Codecs (EVRC) and Selectable Mode Vocoders (SMV)* are supported.

When using EVRCs, more than one codec data frame may be included in a single interleaved/bundled packet by a sender. This is accomplished by:

- Bundling - A technique used to spread the transmission overhead of the RTP and payload header over multiple vocoder frames.
- Interleaving - A technique used to reduce the listener's perception of data loss by spreading such a loss over non-consecutive vocoder frames.

EVRC and similar vocoders can compensate for an occasional lost frame, but speech quality degrades exponentially with consecutive frame loss.

EVRC RTP packet format is controlled by specifying the number of frames per packet (fpp) in the IPM_AUDIO_CODER_INFO structure. Frames per packet values of 1 to 3 are allowed.

When fpp is set to 1, the Header-Free RTP packet format is used.

When fpp is set to 2 or more, the Interleaved/Bundled RTP packet format is used.

Bundling is specified using the unFramesPerPkt field in the IPM_AUDIO_CODER_INFO structure, setting it to 2 or 3.

Interleaving is specified using the unParm1 field in the IPM_AUDIO_CODER_OPTIONS_INFO structure. Valid values are from 0 to 7; 0 indicates do not use interleaving. Interleaving should only be used when the frames per packet value is 2 or more.

Resource reservation of audio coders

> The following information should be added to the programming guide, possibly as a new chapter.

Resource Reservation of Audio Coders

The following sections describe the use of the IPML API and Device Management API in various scenarios for resource reservation of audio coders.

Outbound Call

1. Application reserves coders (RESOURCE_IPM_G711_20MS, RESOURCE_IPM_G723, RESOURCE_IPM_G726, RESOURCE_IPM_G729) and offers the reserved coders to the remote side.

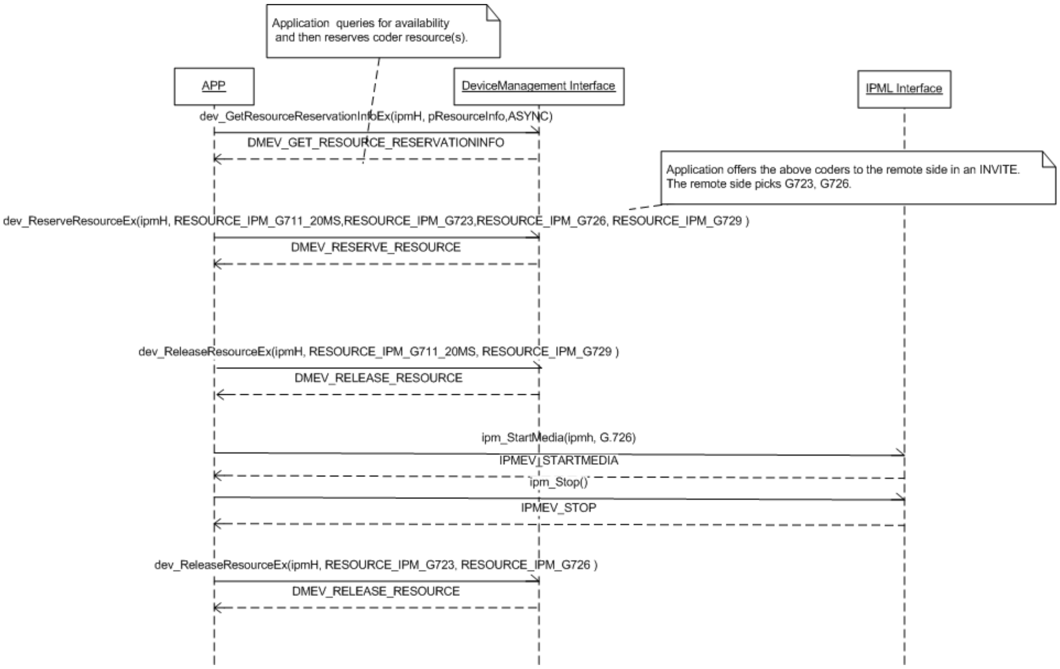
2. The remote side picks G723, G726.

3. Application releases (RESOURCE_IPM_G711_20MS, RESOURCE_IPM_G729). This leaves (RESOURCE_IPM_G723, RESOURCE_IPM_G726) as the reserved audio coders.

4. The application proceeds with the call.

5. After the call is done, the application releases the coders (RESOURCE_IPM_G723, RESOURCE_IPM_G726). (Alternatively, the application can release all the reserved coders by

specifying the enum RESOURCE_IPM_ALL_AUDIO_CODERS.)


Application queries for availability
and then reserves coder resource(s).
APP
DeviceManagement Interface
IPML Interface
dev_GetResourceReservationInfoEx(ipmH, pResourceInfo,ASYNC)
DMEV_GET_RESOURCE_RESERVATIONINFO
Application offers the above coders to the remote side in an INVITE.
The remote side picks G723, G726.
dev_ReserveResourceEx(ipmH, RESOURCE_IPM_G711_20MS,RESOURCE_IPM_G723,RESOURCE_IPM_G726, RESOURCE_IPM_G729)
DMEV_RESERVE_RESOURCE
dev_ReleaseResourceEx(ipmH, RESOURCE_IPM_G711_20MS, RESOURCE_IPM_G729)
DMEV_RELEASE_RESOURCE
ipm_StartMedia(ipmh, G.726)
IPMEV_STARTMEDIA
ipm_Stop()
IPMEV_STOP
dev_ReleaseResourceEx(ipmH, RESOURCE_IPM_G723, RESOURCE_IPM_G726)
DMEV_RELEASE_RESOURCE

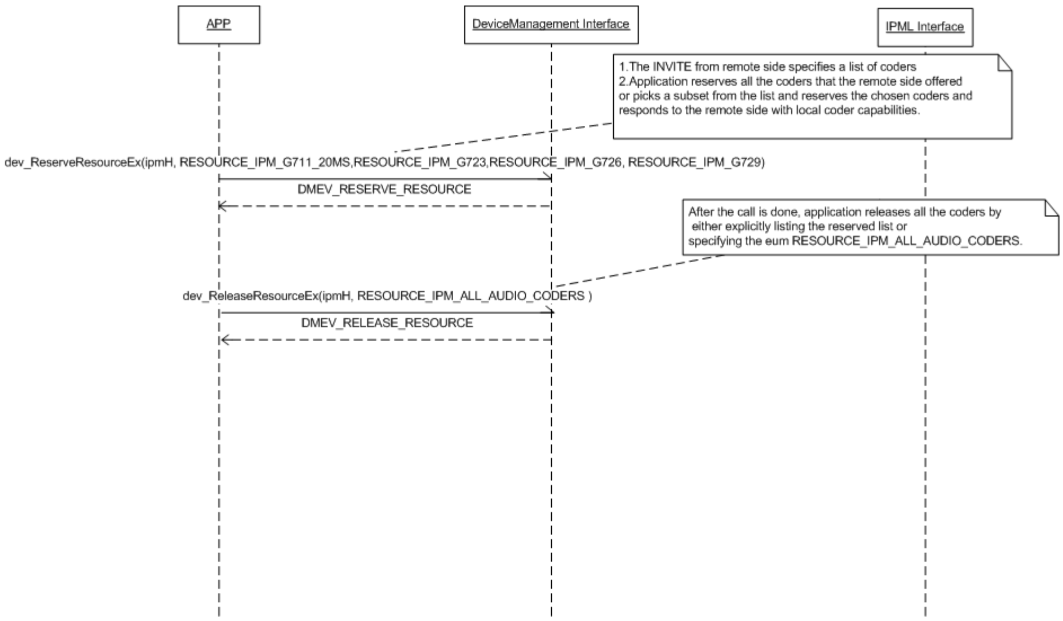
Inbound Call

1. The INVITE from remote side specifies a list of coders.

2. Application may choose to reserve all or some of the coders that the remote side offered and then respond to the remote side with local coder capabilities based on the reservation.

3. Application proceeds with the call.

4. After the call is disconnected, the application releases all the reserved coders by either explicitly listing the reserved list or specifying the enum RESOURCE_IPM_ALL_AUDIO_CODERS.


APP
DeviceManagement Interface
IPML Interface
1.The INVITE from remote side specifies a list of coders
2.Application reserves all the coders that the remote side offered
or picks a subset from the list and reserves the chosen coders and
responds to the remote side with local coder capabilities.
dev_ReserveResourceEx(ipmH, RESOURCE_IPM_G711_20MS,RESOURCE_IPM_G723,RESOURCE_IPM_G726, RESOURCE_IPM_G729)
DMEV_RESERVE_RESOURCE
After the call is done, application releases all the coders by
either explicitly listing the reserved list or
specifying the eum RESOURCE_IPM_ALL_AUDIO_CODERS.
dev_ReleaseResourceEx(ipmH, RESOURCE_IPM_ALL_AUDIO_CODERS)
DMEV_RELEASE_RESOURCE

Implicit Release by a Subsequent Successful Reserve Call

1. Application reserves (RESOURCE_IPM_G726, RESOURCE_IPM_G729).

2. Application uses RESOURCE_IPM_G726 for a media operation (ipm_StartMedia).

3. Application gets a REINVITE from the remote side requesting a change of coder.

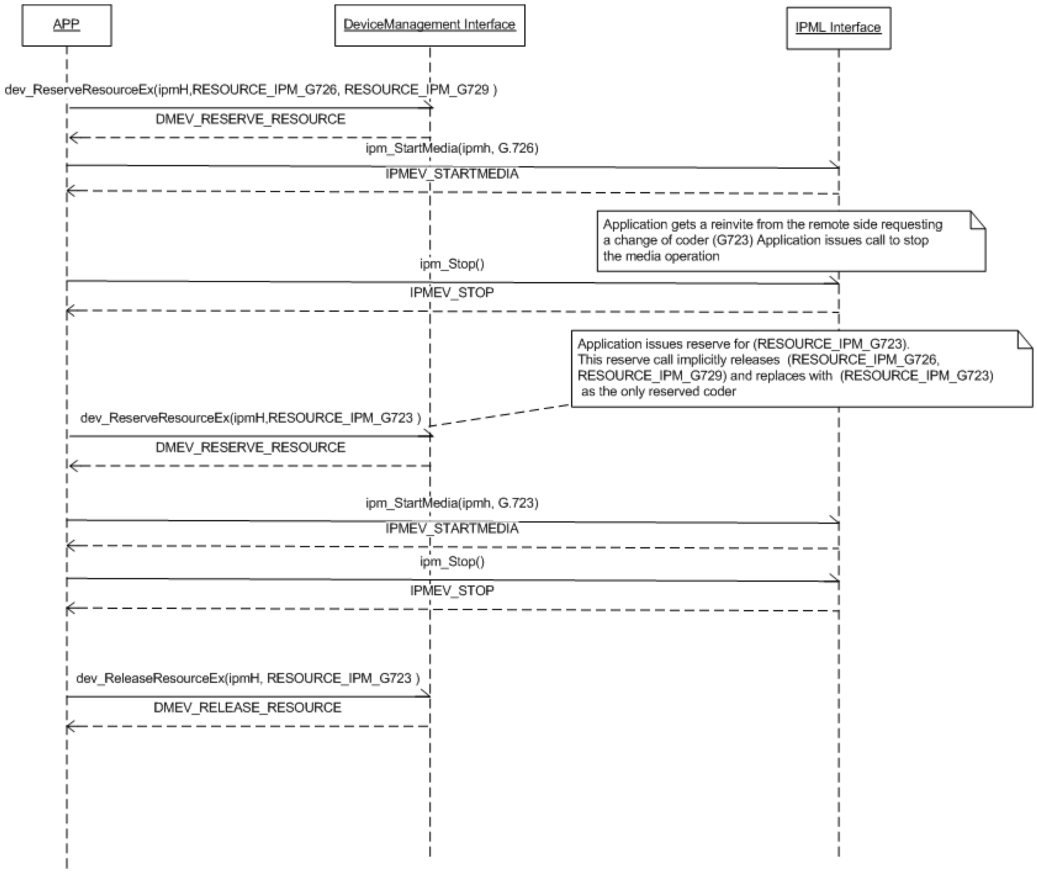
4. Application issues call to stop the media operation.

5. Application issues reserve for RESOURCE_IPM_G723. This reserve call implicitly releases (RESOURCE_IPM_G726,RESOURCE_IPM_G729) and replaces with RESOURCE_IPM_G723 as the only reserved coder.

6. Application uses RESOURCE_IPM_G723 for a media operation (ipm_StartMedia).

7. Application issues call to stop the media operation.

8. Application releases coder by issuing a call to release RESOURCE_IPM_G723.


APP
DeviceManagement Interface
IPML Interface
dev_ReserveResourceEx(ipmH,RESOURCE_IPM_G726, RESOURCE_IPM_G729)
DMEV_RESERVE_RESOURCE
ipm_StartMedia(ipmh, G.726)
IPMEV_STARTMEDIA
Application gets a reinvite from the remote side requesting a change of coder (G723) Application issues call to stop the media operation
ipm_Stop()
IPMEV_STOP
Application issues reserve for (RESOURCE_IPM_G723). This reserve call implicitly releases (RESOURCE_IPM_G726, RESOURCE_IPM_G729) and replaces with (RESOURCE_IPM_G723) as the only reserved coder
dev_ReserveResourceEx(ipmH,RESOURCE_IPM_G723)
DMEV_RESERVE_RESOURCE
ipm_StartMedia(ipmh, G.723)
IPMEV_STARTMEDIA
ipm_Stop()
IPMEV_STOP
dev_ReleaseResourceEx(ipmH, RESOURCE_IPM_G723)
DMEV_RELEASE_RESOURCE

Handling a Resource Reservation Failure

1. A call to reserve resources (RESOURCE_IPM_G726, RESOURCE_IPM_G729) fails for lack of available resources.

Note: The reserve fails when one or more of the resources requested is not available. The application can query to check on resource availability prior to issuing a reserve request. Otherwise, the application will need to query after the reserve fails to identify available/unavailable resources prior to re-issuing a reserve request.

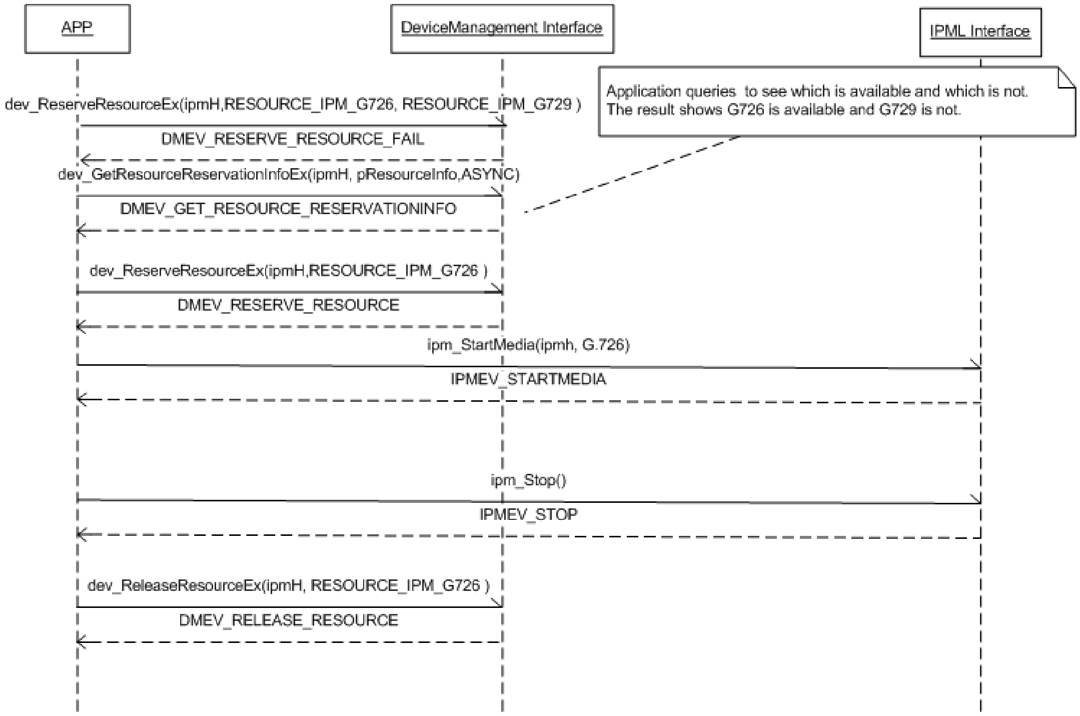
2. Application queries to check on resource availability (for this example, assume this returns RESOURCE_IPM_G726 as available).

3. Application reserves resource RESOURCE_IPM_G726.

4. Application starts media operation (ipm_StartMedia).

5. Application stops media operation (ipm_Stop).

6. Application issues call to release resource RESOURCE_IPM_G726.


APP
DeviceManagement Interface
IPML Interface
Application queries to see which is available and which is not.
The result shows G726 is available and G729 is not.
dev_ReserveResourceEx(ipmH,RESOURCE_IPM_G726, RESOURCE_IPM_G729)
DMEV_RESERVE_RESOURCE_FAIL
dev_GetResourceReservationInfoEx(ipmH, pResourceInfo,ASYNC)
DMEV_GET_RESOURCE_RESERVATIONINFO
dev_ReserveResourceEx(ipmH,RESOURCE_IPM_G726)
DMEV_RESERVE_RESOURCE
ipm_StartMedia(ipmh, G.726)
IPMEV_STARTMEDIA
ipm_Stop()
IPMEV_STOP
dev_ReleaseResourceEx(ipmH, RESOURCE_IPM_G726)
DMEV_RELEASE_RESOURCE

Reservation State after a Subsequent Call to Resource Reservation Fails

1. Application issues call to reserve resources (RESOURCE_IPM_G726, RESOURCE_IPM_G729) and it succeeds.

2. Application starts media operation using G726.

3. Application receives a REINVITE request to use AMR coder.

4. Application stops media operation.

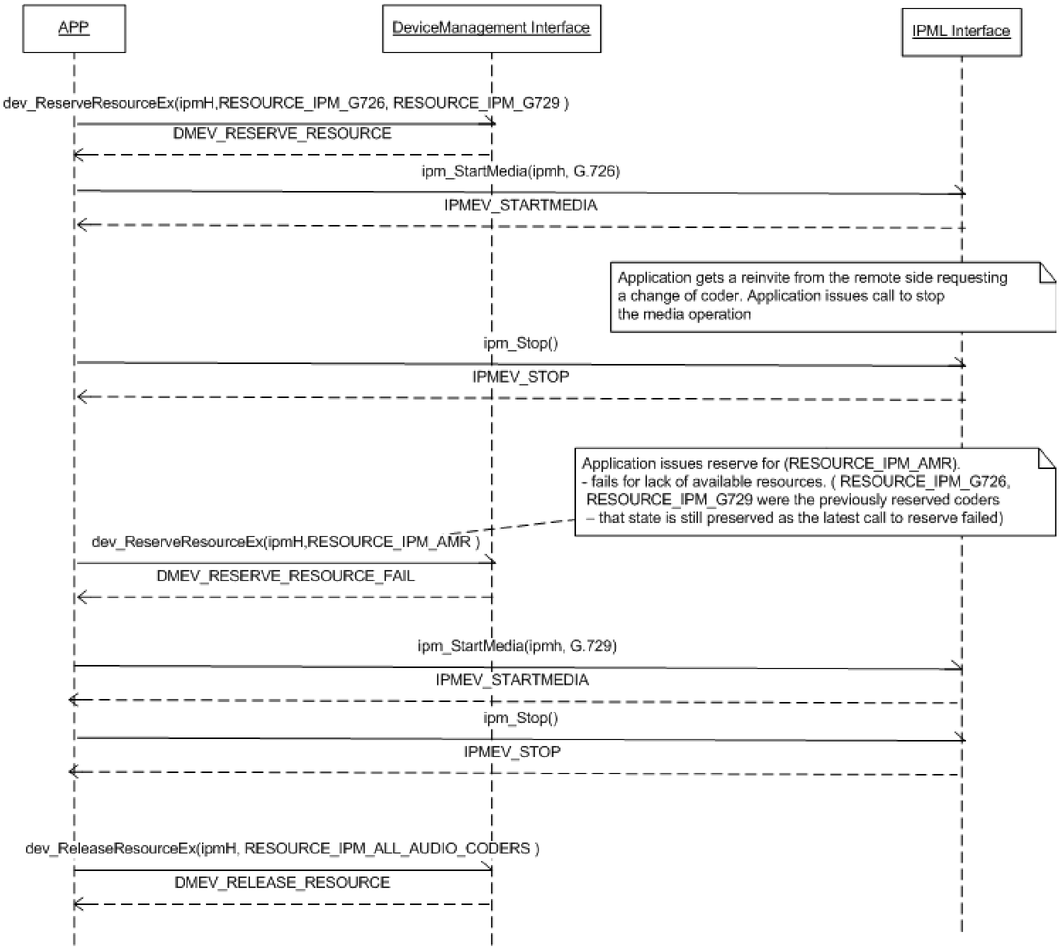
5. Application attempts to reserve RESOURCE_IPM_AMR – the call fails for lack of available resources. The previous successful reserved state is preserved. In this case, (RESOURCE_IPM_G726, RESOURCE_IPM_G729) were the previously reserved coders – that state is still preserved as the latest call to reserve failed.

6. [Application may reject the REINVITE from the remote side and continue to use any of the reserved coders at this point]

7. Start Media Operation.

8. Stop Media Operation.

9. Application issues call to release resources (RESOURCE_IPM_G726, RESOURCE_IPM_G729). It may also use a single enum RESOURCE_IPM_ALL_AUDIO_CODERS to release all the reserved audio coders as opposed to having to list every single one of them.


APP
DeviceManagement Interface
IPML Interface
dev_ReserveResourceEx(ipmH,RESOURCE_IPM_G726, RESOURCE_IPM_G729)
DMEV_RESERVE_RESOURCE
ipm_StartMedia(ipmh, G.726)
IPMEV_STARTMEDIA
Application gets a reinvite from the remote side requesting a change of coder. Application issues call to stop the media operation
ipm_Stop()
IPMEV_STOP
Application issues reserve for (RESOURCE_IPM_AMR).
- fails for lack of available resources. (RESOURCE_IPM_G726, RESOURCE_IPM_G729 were the previously reserved coders – that state is still preserved as the latest call to reserve failed)
dev_ReserveResourceEx(ipmH,RESOURCE_IPM_AMR)
DMEV_RESERVE_RESOURCE_FAIL
ipm_StartMedia(ipmh, G.729)
IPMEV_STARTMEDIA
ipm_Stop()
IPMEV_STOP
dev_ReleaseResourceEx(ipmH, RESOURCE_IPM_ALL_AUDIO_CODERS)
DMEV_RELEASE_RESOURCE

Update to Chapter 13, Building Applications.

Added the following sentence to Section 13.1, Compiling and Linking: When compiling an application, you must list Dialogic libraries before all other libraries such as operating system libraries.

3.4.15 Dialogic® Media Toolkit API Library Reference

There are currently no updates to this document.

3.4.16 Dialogic® Multimedia API Programming Guide and Library Reference

There are currently no updates to this document.

Note: A new version of the document (05-2454-007) is now available on the documentation bookshelf. See the Revision History section of the document for a description of the changes.

3.4.17 Dialogic® Standard Runtime Library API Library Reference

There are currently no updates to this document.

Note: A new document (05-1882-006) is now available on the documentation bookshelf. See the Revision History section of the document for a description of the changes.

3.4.18 Dialogic® Standard Runtime Library API Programming Guide

There are currently no updates to this document.

3.4.19 Dialogic® Voice API Library Reference

Update to Chapter 4, DX_XPB data structure.

The following should be added to the “Field Descriptions” section under wDataFormat:

- DATA_FORMAT_GSM610_MICROSOFT – GSM 6.10 full-rate coder (Microsoft Windows compatible format) (Microsoft Windows Media Recorder Audio Compression Codec: GSM 6.10 Audio CODEC)
- DATA_FORMAT_GSM610_TIPHON – GSM 6.10 VOX full-rate coder (TIPHON format)

The following table should be added to the “Examples” section:

GSM Voice Coder Support Fields

DX_XPB Field	DX_XPB Field Value	Note
wFileFormat	FILE_FORMAT_WAV FILE_FORMAT_VOX	WAVE format supported only with DATA_FORMAT_GSM610_MICROSOFT
wDataFormat	DATA_FORMAT_GSM610_MICROSOFT DATA_FORMAT_GSM610_TIPHON	
nSamplesPerSec	DRT_8KHZ	
wBitsPerSample	0	13 kbps

3.4.20 Dialogic® Voice API Programming Guide

Update to Section 8.5, Voice Encoding Methods.

The following should be added to the “Table 6, Voice Encoding Methods” section:

- GSM 6.10 full rate (Microsoft format)
- GSM 6.10 full rate (TIPHON format)

Update to Chapter 11, Building Applications.

Added the following sentence to Section 11.2, Compiling and Linking: When compiling an application, you must list Dialogic libraries before all other libraries such as operating system libraries.

3.5 Remote Control Interfaces Documentation

This section contains updates to the following documents (click the title to jump to the corresponding section):

- Dialogic® MSML Media Server Software User’s Guide

3.5.1 Dialogic® MSML Media Server Software User’s Guide

There are currently no updates to this document.

3.6 Demonstration Software Documentation

This section contains updates to the following documents (click the title to jump to the corresponding section):

- Dialogic® 3G-324M Multimedia Gateway Demo Guide
- Dialogic® Global Call API Demo Guide
- Dialogic® Multimedia Demo Guide

3.6.1 Dialogic® 3G-324M Multimedia Gateway Demo Guide

There are currently no updates to this document.

3.6.2 Dialogic® Global Call API Demo Guide

There are currently no updates to this document.

3.6.3 Dialogic® Multimedia Demo Guide

There are currently no updates to this document.