



Dialogic® System Release 6.1 for Linux

Release Update

December 2, 2009

Copyright and Legal Notice

Copyright © 2005-2009, 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 furnished for informational use only and are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation or its subsidiaries ("Dialogic"). Reasonable effort is made to ensure the accuracy of the information contained in the document. However, Dialogic does not warrant the accuracy of this information and cannot accept responsibility for errors, inaccuracies 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.

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 may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic does 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 and no such licenses are provided except pursuant to a signed agreement with Dialogic. More detailed information about such intellectual property is available from Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. **Dialogic 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, Diva, Cantata, SnowShore, Eicon, Eicon Networks, NMS Communications, NMS (stylized), Eiconcard, SIPcontrol, Diva ISDN, TruFax, Exnet, EXS, SwitchKit, N20, Making Innovation Thrive, Connecting to Growth, Video is the New Voice, Fusion, Vision, PacketMedia, NaturalAccess, NaturalCallControl, NaturalConference, NaturalFax and Shiva, among others as well as related logos, are either registered trademarks or 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.

Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries. Other names of actual companies and product 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 <http://www.voiceage.com/licensing.php>.

Publication Date: December 2, 2009

Document Number: 05-2466-064

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 Dialogic® System Release 6.1 for Linux (sometimes also referred to herein as “System Release 6.1 Linux”). In addition to summarizing issues that were known as of the Release’s general availability, it is intended that this Release Update will continue to be updated to serve as the primary mechanism for communicating new issues, if any, that may arise after the release date.

Intended Audience

This Release Update is intended for users of System Release 6.1 Linux.

How to Use This Publication

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

- [Document Revision History](#): This section summarizes the ongoing changes and additions that are 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 system release subsequent to the general availability release date. For example, the new features provided in Service Updates are described here.
- [Release Issues](#): This section lists issues that may affect the system release hardware and software. The primary list is sorted by issue type, but alternate sorts by defect number, by product or component, and by Service Update number are also provided.
- [Documentation Updates](#): This section contains corrections and other changes that apply to the System Release documentation set that were not made to the documents prior to the release. The 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® System Release 6.1 for Linux 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/support/> (for Dialogic technical support)
- <http://www.dialogic.com/> (for Dialogic® product information)

Document Revision History

This Revision History summarizes the changes made in each published version of the Release Update for Dialogic® System Release 6.1 for Linux, which is a document that has been and is intended to be periodically updated throughout the lifetime of the release.

Document Rev 64 - published December 2, 2009

Additional Update for Service Update 317.

In the Post-Release Developments section:

- Added [Support for Dialogic® D/4PCIUFEQ and Dialogic® D/4PCIU4SEQ Media Boards](#). Support was added in Service Update 316.

Document Rev 63 - published November 20, 2009

Updated for Service Update 317.

In the Post-Release Developments section:

- Removed Red Flag DC Server 5.0 from [Section 1.37, “Additional Supported Operating System Distributions \(OSDs\)”](#), on page 90. Red Flag DC Server 5.0 is no longer supported. Also added Enterprise Server to Red Hat Enterprise Linux Version 4.0 Update 5.

In the Release Issues section:

- Added the following resolved problems: IPY00081061, IPY00081390.
- Added the following known issue: IPY00080895.

In the Documentation Updates section:

- Added that the Global Call SS7 binaries are now linked with the shared library of the Dialogic® SS7 DSI Development Package in the [Section 3.1.1, “Dialogic® System Release 6.1 for Linux Release Guide”](#), on page 158. (IPY00081381).
 - This change requires that Global Call SS7 customers use the Dialogic® SS7 DSI Development Package version 5.0 or later. If a customer is using an older version, the Global Call SS7 server will not start during download.

Document Rev 62 - published September 30, 2009

Updated for Service Update 316.

In the Post-Release Developments section:

- Added [Support for Dialogic® D/80PCIE-LS Media Board](#).

Document Rev 61 - published June 23, 2009

Updated for Service Update 315.

In the Release Issues section, added the following resolved problems: IPY00074292, IPY00079108, IPY00079561, IPY00079648, IPY00079651, IPY00079668, IPY00079691, IPY00079716, IPY00079825, IPY00079866, IPY00080145, IPY00080244, IPY00080252.

Document Rev 60 - published April 30, 2009

Updated for Service Update 312.

In the Post-Release Developments section:

- Added [Time Stamp and Frame Energy for TDX_CST Event](#).

In the Release Issues section, added the following resolved problems: IPY00079678, IPY00079703, IPY00079797.

Document Rev 59 - published March 31, 2009

Updated for Service Update 311.

In the Release Issues section, added the following resolved problems: IPY00079393, IPY00079399, IPY00079477, IPY00079523.

Document Rev 58 - published February 19, 2009

Updated for Service Update 309.

In the Release Issues section, added the following resolved problems: IPY00078576, IPY00079212, IPY00079213, IPY00079251, IPY00079345, IPY00079365.

In the Documentation Updates section:

- Added an update to the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* to indicate that some coders are not supported on the Dialogic® DM/V4800BC Board with certain media loads.
- Added an update to the *Dialogic® Global Call Country Dependent Parameters (CDP) for PDK Protocols Configuration Guide* for a new parameter in the Brazil R2 Bidirectional protocol *pdk_br_r2_io.cdp* file.
- Added an update to the *Dialogic® Global Call IP Technology Guide* to indicate that the INFO method is included as part of the Allow header in SIP messages by default.
- Deleted the corrections for the *Dialogic® Fax Software Reference*, because these corrections have been incorporated into an updated document that is now on the online documentation bookshelf.

Document Rev 57 - published January 9, 2009

Updated for Service Update 308.

In the Release Issues section, added the following resolved problems: IPY00079095, IPY00079160, IPY00079206.

In the Documentation Updates section:

- Added an update to the *Dialogic® Voice API Library Reference* for the **dx_deltones()** function (IPY00079097).
- Deleted the corrections for the *Dialogic® Digital Network Interface Software Reference*, *Dialogic® Global Call ISDN Technology Guide*, and *Dialogic® PBX Integration Software Reference*, because these corrections have been incorporated into updated documents that are now on the online documentation bookshelf.

Document Rev 56 - published December 9, 2008

Updated for Service Update 307.

In the Post-Release Developments section:

- Added Support for SFTP in *Dialogic® Global Call SS7 Call Control Library*.
- Under Additional Supported Operating System Distributions (OSDs), added a note that only 32-bit versions of Linux OSDs are supported.

In the Release Issues section:

- Added the following resolved problems: IPY00045503, IPY00078799.
- Added the following known problem: IPY00079022.

In the Documentation Updates section:

- Added an update to the *Dialogic® System Release 6.1 for Linux Release Guide*, Basic Software Requirements section, to specify that only 32-bit versions of Linux OSDs are supported.
- Added updates to the *Dialogic® Voice API Library Reference* for the **dx_setsvcond()** function and DX_SVCB data structure (IPY00079103).
- Deleted the corrections for the *Dialogic® Global Call SS7 Technology Guide*, because these corrections have been incorporated into an updated document that is now on the online documentation bookshelf.

Document Rev 55 - published November 18, 2008

Updated for Service Update 304.

In the Release Issues section, added the following resolved problems: IPY00045074, IPY00045478, IPY00078411, IPY00078445, IPY00078519.

In the Documentation Updates section, deleted the corrections for the *Dialogic® Global Call E1/T1 CAS/R2 Technology Guide*, because these corrections have been incorporated into an updated document that is now on the online documentation bookshelf.

Document Rev 54 - published October 22, 2008

Updated for Service Update 301.

In the Release Issues section, added the following resolved problems: IPY00045293, IPY00045376, IPY00045388, IPY00045440, IPY00045442.

In the Documentation Updates section:

- Added an update to the *Dialogic® Continuous Speech Processing API Library Reference* for the **ec_reciottdata()** and **ec_stream()** functions.
- Added an update to the *Dialogic® Voice API Programming Guide*, Application Development Guidelines chapter, regarding continuous speech processing (CSP) resource sharing between multiple processes.
- Added an update to the *Dialogic® Voice API Library Reference* for the **dx_rec()**, **dx_reciottdata()**, **dx_recvox()**, and **dx_recwav()** functions.

Document Rev 53 - published September 30, 2008

Updated for Service Update 300.

In the Post-Release Developments section, added Red Hat Enterprise Linux 5 Update 2 under Additional Supported Operating System Distributions (OSDs).

In the Release Issues section, added the following resolved problems: IPY00045159, IPY00045184, IPY00045224, IPY00045239, IPY00045277, IPY00045343.

In the Documentation Updates section, deleted the corrections for the *Dialogic® Global Call Analog Technology Guide*, because these corrections have been incorporated into an updated document that is now on the online documentation bookshelf.

Document Rev 52 - published September 10, 2008

Updated for Service Update 298.

In the Release Issues section, added the following resolved problems: IPY00044100, IPY00044251, IPY00044425, IPY00045021, IPY00045132, IPY00045136.

In the Documentation Updates section, added an update to the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* for the ISDN **ProtocolType** parameter (IPY00045267).

Document Rev 51 - published September 2, 2008

Updated for Service Update 297.

In the Release Issues section, added the following resolved problems: IPY00044561, IPY00044614, IPY00044811, IPY00044832, IPY00044932. Also revised the information for IPY00038866 - it was resolved in Service Update 297, not in Service Update 273.

In the Documentation Updates section, added an update to the *Dialogic® Standard Runtime Library API Library Reference* for the **sr_getfdcnt()** and **sr_getfdinfo()** functions (IPY00045054).

Document Rev 50 - published August 19, 2008

Updated for Service Update 296.

In the Post-Release Developments section, under Additional Supported Operating System Distributions (OSDs):

- Added SUSE Linux Enterprise Server 9 SP4 and SUSE Linux Enterprise Server 10 SP2.
- For SUSE Linux Enterprise Server 10 and SP1, removed the restriction that only Dialogic® Springware Boards are supported; Dialogic® DM3 Boards are now supported as well. Also removed the restriction on using the Global Call SS7 Support Software package; this package may now be installed when using SUSE Linux Enterprise Server 10 and SP1.

In the Release Issues section, added the following resolved problems: IPY00044544, IPY00044700, IPY00044779.

In the Documentation Updates section:

- Added an update to the *Dialogic® System Release 6.1 for Linux Administration Guide* for the listboards utility (IPY00045020).
- Added an update to the *Dialogic® Voice API Library Reference* for the **dx_OpenStreamBuffer()** function (IPY00044981).
- Deleted the corrections for the *Dialogic® PBX Integration Board User's Guide*, because these corrections have been incorporated into an updated document that is now on the online documentation bookshelf.

Document Rev 49 - published July 22, 2008

Updated for Service Update 294.

In the Release Issues section, added the following resolved problems: IPY00043307, IPY00043801, IPY00044199, IPY00044257, IPY00044273, IPY00044363, IPY00044432.

In the Documentation Updates section, made a correction to the sample code for **gc_GetFrame()** in the *Dialogic® Global Call API Library Reference*.

Document Rev 48 - published July 10, 2008

Updated for Service Update 293.

In the Release Issues section, added the following resolved problems: IPY00043965, IPY00044132, IPY00044185.

In the Documentation Updates section, made a correction to the *Dialogic® Global Call SS7 Technology Guide* about using dual resilient SIU configurations.

Document Rev 47 - published June 19, 2008

Updated for Service Update 292.

In the Post-Release Developments section, added Raw Data Mode Support with Dialogic® DM3 Boards.

In the Release Issues section, added the following resolved problems: IPY00042860, IPY00043818, IPY00043907, IPY00044087.

In the Documentation Updates section:

- Added documentation updates to the *Dialogic® Continuous Speech Processing API Library Reference* and *Dialogic® Voice API Library Reference* because of the new feature in the Service Update for raw data mode support.
- Added an update to the *Dialogic® Global Call API Library Reference* for the **gc_util_insert_parm_val()** function (IPY00043078).
- Added two updates to the *Dialogic® Global Call SS7 Technology Guide*, one about Bearer Independent Call Control (BICC) signaling protocol not supported, and one about opening trunk devices for SS7.
- Deleted the corrections for the *Dialogic® Modular Station Interface API Library Reference*, because these corrections have been incorporated into an updated document that is now on the online documentation bookshelf.

Document Rev 46 - published June 3, 2008

Updated for Service Update 291.

In the Release Issues section, added the following resolved problems: IPY00043609, IPY00043701.

In the Documentation Updates section, deleted the corrections for the *Dialogic® Modular Station Interface API Programming Guide*, because these corrections have been incorporated into an updated document that is now on the online documentation bookshelf.

Document Rev 45 - published May 28, 2008

Updated for Service Update 290.

In the Release Issues section, added the following resolved problems: IPY00043077, IPY00043267, IPY00043292, IPY00043515, IPY00043545.

In the Documentation Updates section, added that a new version of the *Dialogic® Standard Runtime Library API Library Reference* is now available on the online documentation bookshelf.

Document Rev 44 - published May 20, 2008

Updated for Service Update 289.

In the Release Issues section, added the following resolved problems: IPY00043029, IPY00043230, IPY00043240, IPY00043314, IPY00043430, IPY00043432, IPY00043443.

Document Rev 43 - published May 9, 2008

Updated for Service Update 288.

In the Release Issues section:

- Added the following resolved problems: IPY00041808, IPY00043275.
- Added the following known (permanent) problem: IPY00041217.

Document Rev 42 - published April 25, 2008

Updated for Service Update 287.

In the Release Issues section, added the following resolved problems: IPY00042336, IPY00042464, IPY00042579, IPY00042730, IPY00042752, IPY00042828, IPY00042845, IPY00042862, IPY00042940, IPY00042985

In the Documentation Updates section, deleted the corrections for the *Dialogic® Global Call Country Dependent Parameters (CDP) for PDK Protocols Configuration Guide*, because these corrections have been incorporated into an updated document that is now on the online documentation bookshelf.

Document Rev 41 - published April 4, 2008

Updated for Service Update 284.

In the Post-Release Developments section:

- Added File Management Enhancements for DebugAngel Tool.
- Added an option for the Telecom Subsystem Summary Tool (its_sysinfo) to disable collection of board memory dumps.

In the Release Issues section, added the following resolved problems: IPY00040536, IPY00040902, IPY00042204, IPY00042300, IPY00042408, IPY00042528, IPY00042601, IPY00042609.

In the Documentation Updates section, added documentation updates to the *Dialogic® System Software Diagnostics Guide* because of new features in the Service Update.

Document Rev 40 - published March 20, 2008

Updated for Service Update 283.

In the Post-Release Developments section, added Red Hat Enterprise Linux Version 4.0 Update 5 (Advanced Server) and Update 6 (Advanced Server) under Additional Supported Operating System Distributions (OSDs).

In the Release Issues section, added the following resolved problems: IPY00041855, IPY00041959, IPY00041987, IPY00042003, IPY00042168, IPY00042299, IPY00042650.

In the Documentation Updates section, added an update for the *Dialogic® System Release 6.1 for Linux Software Installation Guide* regarding a procedure applicable to SUSE Linux Enterprise Server OSDs.

Document Rev 39 - published February 27, 2008

Updated for Service Update 280.

In the Post-Release Developments section, added New Media Loads for Dialogic® DMV2400A-cPCI and DMV4800BC Media Boards.

In the Release Issues section, added the following resolved problems: IPY00041079, IPY00041209, IPY00041407, IPY00041421, IPY00041580, IPY00041740, IPY00042208.

Document Rev 38 - published February 15, 2008

Updated for Service Update 279.

In the Post-Release Developments section, revised the PBX Integration Support for Nortel BCM section to indicate that the Message Waiting Indication (MWI) and Calling Party ID (CPID) features are now supported. Programming instructions have also been updated, and called/calling number ID data has been added.

In the Release Issues section, added the following resolved problems: IPY00037905, IPY00041792, IPY00041983.

In the Documentation Updates section, added updates for the *Dialogic® PBX Integration Software Reference*.

Document Rev 37 - published February 1, 2008

Updated for Service Update 278.

In the Post-Release Developments section:

- Added Support for SIP re-INVITE on Dialogic® DM/IP Boards.
- In the Support for PCI Express Boards - Dialogic® DM/V-B Boards section, made corrections to the Media Loads table.
- Deleted the detailed descriptions about some Dialogic® Global Call SS7 features that were previously included in this section, because this information has been incorporated into the updated *Dialogic® Global Call SS7 Technology Guide* that is now on the documentation bookshelf.

In the Release Issues section, added the following resolved problems: IPY00039334, IPY00039512, IPY00040179, IPY00041296, IPY00041345, IPY00041369.

In the Documentation Updates section:

- Revised the correction for the **dx_getdig()** function (IPY00038453) under *Dialogic® Voice API Library Reference*.
- Deleted some of the corrections for the *Dialogic® Global Call API Library Reference*, *Dialogic® Global Call IP Technology Guide*, and *Dialogic® Global Call SS7 Technology Guide*, because these corrections have been incorporated into updated documents that are now on the online documentation bookshelf.

Document Rev 36 - published December 28, 2007

Updated for Service Update 276.

In the Release Issues section, added the following resolved problems: IPY00041088, IPY00041111, IPY00041118, IPY00041233, IPY00041300.

Document Rev 35 - published December 14, 2007

Updated for Service Update 275.

In the Post-Release Developments section:

- Added Analog Call Transfer Support on Dialogic® Springware Boards.
- Under Additional Supported Operating System Distributions (OSDs), removed the restriction that only Dialogic® Springware Boards are supported with Red Hat Enterprise Linux 5; Dialogic® DM3 Boards are now supported as well. Added a note about enabling ramdisk if high record failures occur with DM3 CompactPCI Boards.

In the Release Issues section, added the following resolved problems: IPY00040743, IPY00041078, IPY00041129.

In the Documentation Updates section:

- Added documentation updates to the *Dialogic® Global Call Country Dependent Parameters (CDP) for PDK Protocols Configuration Guide*, *Dialogic® Global Call API Library Reference*, and *Dialogic® Global Call Analog Technology User's Guide* because of the new feature in the Service Update for analog call transfer support on Dialogic® Springware Boards.
- Deleted the corrections for the *Dialogic® Audio Conferencing API Programming Guide*, because these corrections have been incorporated into an updated document that is now on the online documentation bookshelf. (Other documents with updated versions on the online documentation bookshelf are: *Dialogic® Audio Conferencing API Library Reference*, *Dialogic® Continuous Speech Processing API Programming Guide*, *Dialogic® Continuous Speech Processing API Library Reference*, *Dialogic® Standard Runtime Library API Programming Guide*, and *Dialogic® Standard Runtime Library API Library Reference*.)

Document Rev 34 - published November 21, 2007

Updated for Service Update 273.

In the Post-Release Developments section:

- Added PBX Integration Support for Nortel BCM.
- Deleted information about open source drivers. (The program is scheduled to be discontinued.)

In the Release Issues section, added the following resolved problems: IPY00038866, IPY00039661, IPY00040685, IPY00040832, IPY00040903.

In the Documentation Updates section:

- Added documentation updates to the *Dialogic® System Release 6.1 for Linux Release Guide* to provide further information about requirements for NetSNMP with Red Hat Enterprise Linux 5 and SUSE Linux Enterprise Server 10 (IPY00041134).
- Added documentation updates to the *Dialogic® Fax Software Reference* for additional return values for **ATFX_RESLN()** and other related changes (IPY00040796).
- Added documentation update to the *Dialogic® PBX Integration Board User's Guide* because of the new feature in the Service Update for Nortel BCM support.

Document Rev 33 - published October 31, 2007

Updated for Service Update 272.

In the Post-Release Developments section, added SUSE Linux Enterprise Server 10 SP 1 under Additional Supported Operating System Distributions (OSDs).

In the Release Issues section, added the following resolved problems: IPY00038391, IPY00039490, IPY00040096, IPY00040439, IPY00040781.

In the Documentation Updates section:

- Added documentation updates to the *Dialogic® Springware Architecture Products on Linux Configuration Guide* and *Dialogic® Continuous Speech Processing API Programming Guide* regarding use of the **EC_Resource** and **CSPEXtraTimeSlot** parameters on Dialogic® Springware Boards (IPY00041018).
- Added documentation update to the *Dialogic® Global Call ISDN Technology Guide* for additional firmware-related cause values when using Dialogic® DM3 Boards (IPY00041046).
- Added information about SIP redirection (3xx) response messages under *Dialogic® Global Call IP Technology Guide*.

Document Rev 32 - published October 19, 2007

Updated for Service Update 271.

In the Post-Release Developments section:

- Added Configuring SIP Stack Parameters with Global Call.
- Added Disabling Automatic re-INVITE Message when Switching between Fax and Audio.
- Under AMD Opteron Server Support, deleted the note about unsupported hardware; the issues have been resolved.
- Added a code example to the IP Multicast Client Support section showing how to start a Multicast client session.
- Under Additional Supported Operating System Distributions (OSDs), revised the information about disabling Advanced Configuration and Power Interface (ACPI) interrupt routing for Red Hat Enterprise Linux 5 and SUSE Linux Enterprise Server 10.

In the Release Issues section, added the following resolved problems: IPY00038551, IPY00039032, IPY00039068, IPY00039155, IPY00039179, IPY00039249, IPY00039401, IPY00039412, IPY00039476, IPY00039539, IPY00039586, IPY00039707, IPY00039847, IPY00039965, IPY00040052.

In the Documentation Updates section:

- Added documentation updates to the *Dialogic® Global Call IP Technology Guide* because of new features in the Service Update.
- Added IPY00006258 (PTR 36353) under *Dialogic® PBX Integration Board User's Guide*.
- Added documentation updates to the *Dialogic® Voice API Programming Guide* and *Dialogic® Voice API Library Reference* for functions that are no longer supported (**r2_creatfsig()** and **r2_playbsig()**).

Document Rev 31 - published September 6, 2007

Additional update for Service Update 268.

In the Post-Release Developments section, added AMD Opteron Server Support.

Document Rev 30 - published August 30, 2007

Updated for Service Update 268.

In the Post-Release Developments section:

- Added IP Multicast Client Support.
- Added Global DPD Enabled on Dialogic® Springware Boards.

In the Release Issues section:

- Added the following resolved problems: IPY00038545, IPY00038981, IPY00039014, IPY00039331, IPY00039341, IPY00039492.
- Eliminated the link to view issues sorted by PTR number. (PTR numbers have been superseded by defect numbers. The PTR numbers still appear in the Release Issues table for historical purposes, but a version of the table sorted by PTR number is no longer provided.)

In the Documentation Updates section, added documentation updates to the following documents because of new features in the Service Update: *Dialogic® IP Media Library API Library Reference*, *Dialogic® Voice API Programming Guide*.

Document Rev 29 - published August 15, 2007

Updated for Service Update 267.

In the Post-Release Developments section, added Setting Data Transfer Buffer Size below 1K for Dialogic® Springware Boards.

In the Release Issues section, added the following resolved problems: IPY00038190, IPY00038433, IPY00038494, IPY00038612, IPY00038919, IPY00038979, IPY00038991, IPY00038998, IPY00039248.

In the Documentation Updates section, added documentation updates to the following documents because of a new feature in the Service Update: *Dialogic® Springware Architecture Products on Linux Configuration Guide* and *Dialogic® Voice API Library Reference*.

Document Rev 28 - published July 10, 2007

Updated for Service Update 262.

In the Post-Release Developments section, updated the Support for PCI Express Boards - Dialogic® Springware Boards section for the Dialogic® D/42JCT-EW and Dialogic® D/82JCT-EW PBX Integration Boards. Also added support for these boards in the New ANI/DNIS-Enabled Parsing Tool (ADEPT) for Dialogic® PBX Integration Boards section.

In the Release Issues section, added the following resolved problems: IPY00037841, IPY00037923, IPY00038130, IPY00038516, IPY00038539, IPY00038572, IPY00038611, IPY00038708, IPY00038836, IPY00038849.

Document Rev 27 - published June 25, 2007

Updated for Service Update 260.

In the Post-Release Developments section:

- Added New Media Load for Dialogic® DMV1200BTEP Media Boards.
- In the Support for PCI Express Boards - Dialogic® DM/V-B Boards section, made minor changes to terminology in the Media Loads table.

In the Release Issues section:

- Added the following resolved problems: IPY00037401, IPY00038060, IPY00038230, IPY00038240, IPY00038244, IPY00038280, IPY00038298, IPY00038365, IPY00038407, IPY00038435, IPY00038477, IPY00038524, IPY00038533. Also added IPY00032797, IPY00037166, and IPY00037861 (resolved in Service Update 257).
- Added a known problem for Dialogic® IPT Boards regarding RTCP and RTP alarms.

In the Documentation Updates section:

- Added updates to the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* about NFAS D channel backup (DCBU) supported on 4ESS, 5ESS, and NI-2.
- Added an update to the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* about active talker and scaling in conferences.

Made global changes to reflect Dialogic brand.

Document Rev 26 - published June 13, 2007

Additional update for Service Update 257.

In the Post-Release Developments section, updated the Support for PCI Express Boards - Springware Products section for the Dialogic® D/240JCT-T1-EW and Dialogic® D/300JCT-E1-EW PCI Express Boards.

Document Rev 25 - published June 4, 2007

In the Post-Release Developments section, added Troubleshooting Information for RTF Logs.

Document Rev 24 - published June 1, 2007

Additional update for Service Update 257.

In the Post-Release Developments section, added Red Hat Enterprise Linux 5 and SUSE Linux Enterprise Server 10 under Additional Supported Operating System Distributions (OSDs).

Note: Refer to the Additional Supported Operating System Distributions (OSDs) section for limitations when using Red Hat Enterprise Linux 5 and SUSE Linux Enterprise Server 10. These limitations will be removed in a future Service Update.

Document Rev 23 - published May 24, 2007

Updated for Service Update 257.

In the Post-Release Developments section:

- Added Support for LiS 2.19.1.
- Added Remote Diagnostics Package.
- Added Enhanced Diagnostics Tools.
- Added Support for PCI Express Boards - DM3 DI Series Products.
- In the Support for PCI Express Boards - DM/V-B Products and Support for PCI Express Boards - Springware Products sections, revised names of the PCI Express Boards to indicate their item market names.
- Updated the New Diagnostics Management Console section to add more tools that can now be executed: Pstndiag and StatusMon.

In the Release Issues section, added the following resolved problems: IPY00030001 (PTR 36796), IPY00034857, IPY00036665, IPY00037183, IPY00037319, IPY00037373, IPY00037396, IPY00037507, IPY00037721, IPY00037796, IPY00037817, IPY00037818, IPY00037918, IPY00038215.

Note: The fix for defect **IPY00037796** may have an impact on existing DM3 applications; refer to the defect description in the Release Issues section.

In the Documentation Updates section:

- Added updates to the *Dialogic® System Release 6.1 for Linux Release Guide* and *Dialogic® System Release 6.1 for Linux Software Installation Guide* for LiS 2.19.1 support.
- Added updates to the *Dialogic® System Software Diagnostics Guide* for the new diagnostics features in the Service Update.
- Added an update for the **gc_InitXfer()** function under *Dialogic® Global Call API Library Reference* (IPY00038401).
- Added an update for the **dx_getdig()** function under *Dialogic® Voice API Library Reference* (IPY00038453).
- Added an update for the **dx_setevtmask()** function under *Dialogic® Voice API Library Reference* (IPY00038053).

Document Rev 22 - published April 24, 2007

Updated for Service Update 251.

In the Post-Release Developments section, updated the Support for PCI Express Boards - Springware Products section for the Dialogic® D/480JCT and Dialogic® D/600JCT PCI Express Boards.

In the Release Issues section, added the following resolved problems: IPY00036830, IPY00036833, IPY00036955, IPY00037002, IPY00037318, IPY00037483, IPY00037607, IPY00037632, IPY00037708, IPY00037767.

Document Rev 21 - published March 30, 2007

Updated for Service Update 248.

In the Post-Release Developments section, added more information about the Dialogic® DMV300BTEPEQ, Dialogic® DMV600BTEPEQ, and Dialogic® DMV1200BTEPEQ Boards under Support for PCI Express Boards - DM/V-B Products. Also corrected the table footnote about echo cancellation in the media loads table.

In the Release Issues section, added the following resolved problems: IPY00034618, IPY00036054, IPY00036856, IPY00036865, IPY00036886, IPY00036919, IPY00037542, IPY00037612, IPY00037633.

In the Documentation Updates section:

- Added information about clock fallback in mixed systems to the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* and *Dialogic® Springware Architecture Products on Linux Configuration Guide* (IPY00036875).
- Added an update to the Media Load table under *Dialogic® DM3 Architecture Products on Linux Configuration Guide*.

Document Rev 20 - published March 19, 2007

Updated for Service Update 247.

In the Release Issues section, added the following resolved problems: IPY00036877, IPY00036920, IPY00036965, IPY00037437.

In the Documentation Updates section, added information about binary log files to the *Dialogic® System Software Diagnostics Guide* (IPY00037518).

Document Rev 19 - published March 13, 2007

In the Post-Release Developments section, updated the Support for PCI Express Boards section for additional PCI Express Boards: Dialogic® D/4PCIU4S, Dialogic® D/4PCIUF, Dialogic® D/41JCT-LS, and Dialogic® VFX/41JCT-LS.

Document Rev 18 - published March 6, 2007

Updated for Service Update 245.

In the Post-Release Developments section:

- Updated the Support for PCI Express Boards section for additional PCI Express Boards: Dialogic® DMV300BTEPEQ, Dialogic® DMV600BTEPEQ, and Dialogic® DMV1200BTEPEQ.
- Added New Parameter for Adjusting Silence Threshold on DM3 Boards.

In the Release Issues section, added IPY00006707 (PTR 33803), IPY00007470 (PTR 32437), IPY00009499 (PTR 33932), IPY00028633 (PTR 35748), IPY00036347, IPY00036423, IPY00036504, IPY00036861.

Document Rev 17 - published February 23, 2007

Updated for Service Update 243.

In the Release Issues section:

- Added the following resolved problems: IPY00006077 (PTR 36327), IPY00006790 (PTR 35137), IPY00033164, IPY00033472, IPY00033763, IPY00034404, IPY00034816, IPY00035148, IPY00035451, IPY00036248, IPY00036280, IPY00036337, IPY00036448.
- Added the following known (permanent) problem: IPY00037015.

Document Rev 16 - published January 22, 2007

Updated for Service Update 241.

In the Post-Release Developments section:

- Added New ANI/DNIS-Enabled Parsing Tool (ADEPT) for Dialogic® PBX Integration Boards.
- Updated the New Diagnostics Management Console section to add more tools that can now be executed: AppMon, Castrace, Isdntrace, Dlgsnapshot, Dm3post, Debugangel, and Pdktrace
- Updated the New Version of its_sysinfo Tool section to add new Linux Package Info that is now included in the its_sysinfo.htm file.

In the Release Issues section, added the following resolved problems: IPY00031534, IPY00032794, IPY00032803, IPY00033185, IPY00034105, IPY00034495, IPY00034678, IPY00035506, IPY00036044, IPY00036247.

In the Documentation Updates section:

- Added a documentation update to the following document because of a new feature in the Service Update: *Dialogic® System Software Diagnostics Guide*.

- Added IPY00006024 (PTR 29612) under *Dialogic® PBX Integration Board User's Guide*.
- Added the *Dialogic® ADEPT for PBX Integration Boards User's Guide* to the bookshelf. Also, added information pertaining to the Linux release.

Document Rev 15 - published January 2, 2007

Updated for Service Update 239.

In the Post-Release Developments section, added Dynamically Changing the Transmit Time Slot on IP Media Devices.

In the Release Issues section, added the following resolved problems: IPY00010760 (PTR 36647), IPY00011037 (PTR 36677), IPY00010929 (PTR 36497), IPY00028222 (PTR 36483), IPY00028460 (PTR 36298), IPY00033563, IPY00033912, IPY00034036, IPY00034406, IPY00034559, IPY00034627, IPY00034841, IPY00035822, IPY00035831.

In the Documentation Updates section, added updates to the *Dialogic® IP Media Library API Programming Guide* and *Dialogic® IP Media Library API Library Reference* for dynamically changing the transmit time slot on IP media devices.

Document Rev 14 - published December 11, 2006

Updated for Service Update 237.

In the Release Issues section, added the following resolved problems: IPY00035860, IPY00036035.

Document Rev 13 - published November 17, 2006

Updated for Service Update 234.

In the Release Issues section:

- Added the following resolved problem: IPY00035660.
- Added the following known problems: IPY00010760 (PTR 36647), IPY00010929 (PTR 36497), IPY00011037 (PTR 36677), IPY00028222 (PTR 36483), IPY00028460 (PTR 36298). These were incorrectly listed as resolved in Service Update 232.

Document Rev 12 - published November 10, 2006

Updated for Service Update 232.

In the Post-Release Developments section:

- Added Runtime Control of Double Answer for R2MF.
- Added Support for Dialogic® D/4PCI Board.

- Added two more supported boards, Dialogic® DM/V2400A and Dialogic® DMV4800BC Media Boards, to Additional Voice Channels on Clear Channel Media Loads.
- Added Support for PCI Express Boards.
- Added updates to Additional Supported Operating System Distributions (OSDs) for Red Hat® Enterprise Linux Versions 3.0 and 4.0.

In the Release Issues section, added the following resolved problems: IPY00010760 (PTR 36647), IPY00010929 (PTR 36497), IPY00011037 (PTR 36677), IPY00028222 (PTR 36483), IPY00028460 (PTR 36298), IPY00029922 (PTR 35353), IPY00033698, IPY00034606, IPY00034738.

Document Rev 11 - published September 5, 2006

Updated for Service Update 226.

Note: The Release Issues section has been modified to show issues by Change Control System defect number and by PTR number.

In the Post-Release Developments section:

- Added Support for 12 GB RAM.
- Added Support for Reporting Billing Type.
- Added Dynamically Adding and Deleting SS7 Circuit Groups.
- Added Global Call Support for Time Slots on Dialogic® SS7 Boards Running in DTI Mode.
- Added Additional Voice Channels on Clear Channel Media Loads.
- Added Media Channel Reset Capability (Stuck Channel Recovery).

In the Documentation Updates section, added a revised statement on system requirements under *Dialogic® System Release 6.1 for Linux Release Guide*.

Document Rev 10 - published August 21, 2006

Updated for Service Update 225.

In the Post-Release Developments section:

- Added Additional Supported Operating System Distributions (OSDs).
- Added New Diagnostics Management Console.
- Added More Configurations for Optional Use of Sharing of Timeslot (SOT) Algorithm.
- Added Ability to Send and Receive DPNSS End to End Messages.
- Added Time Stamp for Tone-On/Off Events.
- Added OA&M Error Cleanup.

Document Rev 09 - published June 29, 2006

Updated for Service Update 217.

In the Post-Release Developments section:

- Added Additional Supported Operating System Distributions (OSDs).
- Added New Runtime Trace Facility (RTF) Manager.
- Added New Application Monitor.
- Added Improved Tracing and Error Reporting.

Document Rev 08 - published June 7, 2006

Updated for Service Update 213.

In the Post-Release Developments section:

- Added New Media Load for Dialogic® DMV600BTEC Boards.
- Added Removal of IP Gateway R4 and IPML Gateway Demos.

In the Release Issues section, added the following resolved problem: 36598.

In the Documentation Updates section, removed *Dialogic® IP Gateway (Global Call) Demo Guide* since the demo is no longer supported.

Document Rev 07 - published April 28, 2006

Updated for Service Update 208.

In the Post-Release Developments section:

- Added New Status Monitor (statusmon) Application.
- Added New Version of Runtime Trace Facility (RTF) Tool.
- Added New Version of Get Version (Getver) Tool.
- Added New Version of Telecom Subsystem Summary (its_sysinfo) Tool.

In the Release Issues section, added the following resolved problems: 35670, 36644, 36670, 36790.

In the Documentation Updates section, removed documentation updates for the *Dialogic® System Software Diagnostics Guide* since these updates have been incorporated into the latest version of the Diagnostics Guide which is now available online.

Document Rev 06 - published April 18, 2006

Updated for Service Update 205.

In the About This Publication section, added related information for the Open System Release website.

In the Post-Release Developments section:

- Added Support for Open Source Drivers.
- Added Optional Use of Sharing of Timeslot (SOT) Algorithm.
- Added Deprecation of Dlgsnmpd and Orbacus Init Scripts.
- Added Removal of Unused Header Files.

In the Release Issues section:

- Added the following resolved problems: 36656, 36657, 36682, 36725, 36727, 36817, 36833.
- Added known problem noting that MLFN media load for 60 channel fax is not supported on Dialogic® DMV600BTEC Board.

Document Rev 05 - published April 5, 2006

Updated for Service Update 204.

In the Release Issues section:

- Added the following resolved problems: 33530, 36452, 36481, 36550.
- Added the following known permanent problem: 35891.
- Added the following known problems: 36581, 36877.

Document Rev 04 - published March 10, 2006

Updated for Service Update 198.

In the Post-Release Developments section:

- Added Additional Supported Operating System Distributions (OSDs).
- Added Dynamic Detection of ETSI FSK Protocols.
- Added **dx_stopch()** EV_NOSTOP Mode Support for Dialogic® DM3 Boards.

In the Release Issues section:

- Added the following resolved problems: 31991, 34433, 35955, 36439, 36519, 36577, 36655.
- Added the following known problems: 35851, 36104, 36452, 36481, 36550, 36564, 36644, 36656, 36657, 36659, 36670, 36682, 36697, 36714, 36727, 36817, 36818.

In the Documentation Updates section:

- Added a new procedure under *Dialogic® System Release 6.1 for Linux Software Installation Guide*.
- Added PTR# 36726 under *Dialogic® Global Call E1/T1 CAS/R2 Technology Guide*.

- Added PTR# 35565 under *Dialogic® Modular Station Interface API Library Reference*.
- Added PTR# 34546 and PTR# 35667 under *Dialogic® Voice API Programming Guide*.
- Added new procedures under *Pigeon Point Systems Linux Hot Swap Kit User Guide*.

Document Rev 03 - published December 13, 2005

Updated for Service Update 189.

In the Post-Release Developments section:

- Added Additional Supported Operating System Distributions (OSDs).
- Added Supported Kernel and GCC Versions.
- Added New Media Loads for Dialogic® DMV1200BTEC Boards.
- Added GSM-AMR-NB Coder Support for Dialogic® IPT Boards.
- Added Tcl/Tk No Longer Required for QScript Utilities Support.
- Added System Logging Integrated with Runtime Trace Facility (RTF).
- Added Support for Multiple Dialogic® SS7 Boards in the Same System.

In the Release Issues section:

- Added the following resolved problems: 35532, 35763, 35819, 35842, 35860, 35861, 35874, 35923, 35925, 36075, 36093, 36123, 36125, 36402, 36408.
- Added the following known permanent problems: 36102, 36155, 36198, 36381.
- Added the following known problems: 36294, 36322, 36394.

In the Documentation Updates section:

- Removed documentation updates for the *Dialogic® System Release 6.1 for Linux Software Installation Guide* since these updates have been incorporated into the latest version of the Software Installation Guide which is now available online.
- Added PTR# 36306 and PTR# 36343 under *Dialogic® System Release 6.1 for Linux Administration Guide*.
- Added correction that Tcl/Tk no longer needs to be installed under *Dialogic® System Release 6.1 for Linux Administration Guide* and *Dialogic® System Software Diagnostics Guide*.
- Added correction that states support for multiple Dialogic® SS7 Boards in the same system under *Dialogic® Global Call SS7 Technology Guide*.
- Added PTR# 35969 under *Dialogic® ISDN Software Reference*.

Document Rev 02 - published October 7, 2005

Updated for Service Update 171.

In the Post-Release Developments section:

- Added Service Update for Dialogic® System Release 6.1 for Linux.

- Added Support for Dialogic® IP Boards and Features.
- Added Support for Redundant Host (RH).
- Added Support for Peripheral Hot Swap (PHS) on Additional Compute Platforms.
- Added New Media Load for Dialogic® DMV4800BC Boards.
- Added Dlgsnapshot Tool's Autodump Feature Now Disabled by Default.

In the Release Issues section:

- Added the following resolved problems: 35554, 35615, 35794, 35821, 35853, 35888.
- Added the following known problems: 36048, 36075, 36093, 36102, 36104, 36123, 36125, 36131.

In the Documentation Updates section:

- Added PTR# 36105 under *Dialogic® System Release 6.1 for Linux Release Guide*.
- Added PTR# 35965 under *Dialogic® Global Call API Library Reference*.

Document Rev 01 - published August 31, 2005

Initial version of document.

Post-Release Developments

This section describes significant changes to the system release subsequent to the general availability release date.

- Service Update for Dialogic® System Release 6.1 for Linux 29
- Support for Dialogic® D/4PCIUFEQ and Dialogic® D/4PCIU4SEQ Media Boards29
- Support for Dialogic® D/80PCIE-LS Media Board 30
- Time Stamp and Frame Energy for TDX_CST Event. 31
- Support for SFTP in Dialogic® Global Call SS7 Call Control Library 36
- Raw Data Mode Support with Dialogic® DM3 Boards 36
- File Management Enhancements for DebugAngel Tool 39
- New Media Loads for Dialogic® DM/V2400A-cPCI and DMV4800BC Media Boards44
- Support for SIP re-INVITE on Dialogic® DM/IP Boards 47
- Analog Call Transfer Support on Dialogic® Springware Boards..... 47
- PBX Integration Support for Nortel BCM 47
- Configuring SIP Stack Parameters with Global Call 48
- Disabling Automatic re-INVITE Message when Switching between Fax and Audio51
- AMD Opteron Server Support. 55
- IP Multicast Client Support 55
- Global DPD Enabled on Dialogic® Springware Boards 57
- Setting Data Transfer Buffer Size below 1K for Dialogic® Springware Boards. 57
- New Media Load for Dialogic® DMV1200BTEP Media Boards 60
- Troubleshooting Information for RTF Logs 61
- Support for LiS 2.19.1 62
- Remote Diagnostics Package. 62
- Enhanced Diagnostics Tools 64
- Support for PCI Express Boards - Dialogic® Station Interface Boards..... 66
- Support for PCI Express Boards - Dialogic® DM/V-B Boards..... 67
- New Parameter for Adjusting Silence Threshold on Dialogic® DM3 Boards .. 70
- New ANI/DNIS-Enabled Parsing Tool (ADEPT) for Dialogic® PBX Integration Boards 70
- Dynamically Changing the Transmit Time Slot on IP Media Devices..... 71
- Runtime Control of Double Answer for R2MF 76

• Support for Dialogic® D/4PCI Voice Board	76
• Support for PCI Express Boards - Dialogic® Springware Boards	77
• Support for 12 GB RAM	79
• Support for Reporting Billing Type	79
• Dynamically Adding and Deleting SS7 Circuit Groups	80
• Dialogic® Global Call API Support for Time Slots on Dialogic® SS7 Boards Running in DTI Mode	80
• Additional Voice Channels on Clear Channel Media Loads	80
• Media Channel Reset Capability (Stuck Channel Recovery)	82
• Additional Supported Operating System Distributions (OSDs)	90
• New Dialogic® Diagnostics Management Console	92
• More Configurations for Optional Use of Sharing of Timeslot (SOT) Algorithm	92
• Ability to Send and Receive DPNSS End to End Messages	93
• Time Stamp for Tone-On/Off Events	93
• OA&M Error Cleanup	95
• New Runtime Trace Facility (RTF) Manager	95
• New Application Monitor	95
• Improved Tracing and Error Reporting	96
• New Media Load for Dialogic® DMV600BTEC Boards	96
• Removal of IP Gateway R4 and IPML Gateway Demos	97
• New Status Monitor (statusmon) Application	97
• New Version of Runtime Trace Facility (RTF) Tool	98
• New Version of Get Version (Getver) Tool	98
• Telecom Subsystem Summary Tool (its_sysinfo)	98
• Optional Use of Sharing of Timeslot (SOT) Algorithm	100
• Deprecation of Dlgcsnmpd and Orbacus Init Scripts	100
• Removal of Unused Header Files	101
• Dynamic Detection of ETSI FSK Protocols	101
• dx_stopch() EV_NOSTOP Mode Support for Dialogic® DM3 Boards	103
• Supported Kernel and GCC Versions	104
• New Media Loads for Dialogic® DMV1200BTEC Boards	104
• GSM-AMR-NB Coder Support for Dialogic® IPT Boards	107
• Tcl/Tk No Longer Required for QScript Utilities Support	108
• System Logging Integrated with Runtime Trace Facility (RTF)	108

- Support for Multiple Dialogic® SS7 Boards in the Same System 109
- Support for Dialogic® DM/IP Boards and Features. 109
- Support for Redundant Host (RH) 110
- Support for Peripheral Hot Swap (PHS) on Additional Compute Platforms . . . 110
- New Media Load for Dialogic® DMV4800BC Boards 111
- Dlgsnapshot Tool Autodump Feature Now Disabled by Default. 113

1.1 Service Update for Dialogic® System Release 6.1 for Linux

A Service Update for Dialogic® System Release 6.1 for Linux is now available. Service Updates provide fixes to known problems, and may also introduce new functionality. New versions of the Service Update are planned to be released periodically. It is intended that this Release Update will document the features in the Service Updates.

Depending on whether you already have a version of Dialogic® System Release 6.1 for Linux on your system, installing the Service Update will give you either a **full install** or an **update install**:

- If you don't have an existing version of System Release 6.1 Linux on your system, installing the Service Update gives you a **full install** of the release. You can select the features that you want to install, for example, Dialogic® DMV/DMN/DMT, Global Call Protocols, Documentation, etc.
- If you have an existing version of System Release 6.1 Linux on your system, installing the Service Update gives you an **update install**. The update install gives you the latest software for the features that you selected when you did the full install of the system release that is currently on your system.

1.2 Support for Dialogic® D/4PCIUFEQ and Dialogic® D/4PCIU4SEQ Media Boards

Service Update 316 provides support for the Dialogic® D/4PCIUFEQ and Dialogic® D/4PCIU4SEQ Media Boards. Both are RoHS commercial product 6/6 half-length PCI Express form factor boards. The boards have the same features and functionality as the current analog Dialogic® JCT boards, except they have no CTBus connectivity.

Note: For more information about RoHS compliance, refer to www.dialogic.com/rohs/default.htm.

The Dialogic® D/4PCIUFEQ provides basic voice processing and DSP-based Group 3 fax support (DSP fax or SoftFax). The Dialogic® D/4PCIU4SEQ supports basic voice processing with continuous speech processing (CSP).

When installing the board(s), be sure to refer to the Installation Guide (Dialogic® Quick Install Card) that is provided with each board for important information about power budgeting and guidelines for selecting the slot where a board can be installed.

1.3 Support for Dialogic® D/80PCIE-LS Media Board

Service Update 316 provides support for the Dialogic® D/80PCIE-LS Media Board. The board is an eight-port analog, PCI Express, loop start board used for developing advanced communications applications that require media resources. The Dialogic® D/80PCIE-LS Media Board has the same features and functionality as the current analog Dialogic® JCT boards.

The board provides support for basic voice processing, DSP-based Group 3 fax support (DSP fax or SoftFax) and continuous speech processing (CSP) in one PCI Express slot.

When installing the Dialogic® D/80PCIE-LS Media Board, be sure to refer to the Installation Guide (Dialogic® Quick Install Card) that is provided with each board for important information about power budgeting and guidelines for selecting the slot where a board can be installed.

Supported Coders

The Dialogic® D/80PCIE-LS Media Board is supported by the following voice encoding methods and sampling rates:

Digitizing Method	Sampling Rate (kHz)	Resolution (Bits)	Bit Rate (Kbps)	File Format
OKI ADPCM	6	4	24	VOX, WAVE
OKI ADPCM	8	4	32	VOX, WAVE
G.711 PCM A-law and mu-law	6	8	48	VOX, WAVE
G.711 PCM A-law and mu-law	8	8	64	VOX, WAVE
Linear PCM	6	8	48	VOX, WAVE
Linear PCM	8	8	64	VOX, WAVE
GSM full rate (Microsoft format)	8	(value ignored)	13	WAVE
GSM full rate (TIPHON format)	8	(value ignored)	13	WAVE
G.726, ITU-T ADPCM	8	4	32	VOX

See the *Dialogic® Voice API Programming Guide* for more information about configuration procedures and firmware load files.

Firmware Load File

The Dialogic® D/80PCIE-LS Media Board requires a firmware load file for the system software to download to the board.

The firmware files available for the Dialogic® D/80PCIE-LS Media Board are as follows:

D8xjct.fwl Provides eight channels of basic voice processing and fax.

D81jcsp.fwl Provides eight channels of basic voice processing, fax, and continuous speech processing (CSP).

Note: *D8xjct.fwl* is the default.

See the *Dialogic® Springware Architecture Products on Linux Configuration Guide* for more information about firmware load files.

1.4 Time Stamp and Frame Energy for TDX_CST Event

Service Update 312 provides a time stamp and frame energy for certain TDX_CST events on the following Dialogic® JCT boards: D/41JCT-LS, D/120JCT-LS, D/240JCT-1T1, D/480JCT-1T1, D/480JCT-2T1, D/300JCT-1E1, D/600JCT-1E1, D/600JCT-2E1 and D/4PCIU.

1.4.1 Feature Description

This feature adds both on-board time stamp and frame energy related information for the TDX_CST GTD Tone event. It also provides time stamp information on silence on/off. The structures, TN_TIMESTAMP and FM_ENERGY, if available for the event, are retrieved upon receiving the TDX_CST event, with a CST event type DE_TONEON, DE_TONEOFF, DE_SILON or DE_SILOFF on Dialogic® JCT boards.

The Voice API library currently supports the TN_TIMESTAMP structure. A new structure, FM_ENERGY, has been added to the *dxxxlib.h* header file to support this feature.

TN_TIMESTAMP for GTD Tone and Silence DX_CST Events

Upon detection of a TDX_CST asynchronous event with tone or silence types (DE_TONEON, DE_TONEOFF, DE_SILON, DE_SILOFF) in the *cst_event* field, the application should retrieve a pointer to the TN_TIMESTAMP structure. The time stamp data is returned in this structure.

- For tone events, the DX_CST, the TN_INFO structure, and the TN_TIMESTAMP are back-to-back to each other on the pointer returned by the **sr_getevtdatap()** function.
- For silence events, the TN_TIMESTAMP follows the DX_CST back-to-back on the pointer returned by the **sr_getevtdatap()** function.

Definitions for the DX_CST, TN_INFO and TN_TIMESTAMP structures are as follows:

```

typedef struct {
    DX_CST cst;
    TN_INFO tninfo;
    TN_TIMESTAMP tnTimeStamp;
} TONEON_INFO;

typedef struct dx_cst {
    unsigned short cst_event;      /* CST Event */
    unsigned short cst_data;      /* Data Associated with the Event */
} DX_CST;

typedef struct {
    unsigned short tn_freq1;      /* Actual Freq of Tone 1 detected (Hz) */
    unsigned short tn_freq2;      /* Actual Freq of Tone 2 detected (Hz) */
    unsigned short tn_on;         /* Actual On Time (in 10msec units) */
    unsigned short tn_off;        /* Actual Off Time (in 10msec units) */
    unsigned short tn_rep_cnt;    /* Actual Rep Count */
    unsigned short rfu;           /* Reserved */
} TN_INFO;

typedef struct {
    unsigned long tn_TimeStamp;
} TN_TIMESTAMP;

```

- Notes:**
1. On a `cst_event = DE_TONEON`, the `TN_INFO` is populated; however, on a `cst_event = DE_TONEOFF`, the `TN_INFO` data is not populated and its data should not be accessed.
 2. The `TN_INFO` structure will not be present for `DE_SILON` and `DE_SILOFF` event types.
 3. The time stamp for a tone on/off event is in milliseconds from when the firmware was downloaded on the board. There is no correlation to the system time, and the time stamp may wrap around after a certain period of time.
 4. If a GTD Tone is detected as both tone and silence, its time stamps will be slightly different. This is because determination of silence and GTD conditions follow different algorithms in the firmware.
 5. This feature is only supported on `TDX_CST` asynchronous event notification. Retrieval of this data structure through the `dx_getevt()` function is not supported.

The following update has been made to the `dxxxlib.h` header file:

```

typedef struct dx_cst {
    unsigned short cst_event;      /* CST Event */
    unsigned short cst_data;      /* Data Associated with the Event */
} DX_CST;

typedef struct {
    unsigned short tn_freq1;      /* Actual Freq of Tone 1 detected (Hz) */
    unsigned short tn_freq2;      /* Actual Freq of Tone 2 detected (Hz) */
    unsigned short tn_on;         /* Actual On Time (in 10msec units) */
    unsigned short tn_off;        /* Actual Off Time (in 10msec units) */
    unsigned short tn_rep_cnt;    /* Actual Rep Count */
    unsigned short rfu;           /* Reserved */
} TN_INFO;

typedef struct {
    unsigned long tn_TimeStamp; /* Time stamp for tone and silence on/off event.
                                * The timestamp will be in milliseconds

```



```

from when
                                                                    */ the firmware was downloaded on the
board.

} TN_TIMESTAMP

```

FM_ENERGY for GTD Tone DX_CST Events

This feature provides the total audio energy level in dBm units of the frame upon DE_TONEON and DE_TONEOFF detection and energy level per frequency peak, also in dBm units. All dBm values are rounded to an integer number.

Upon providing this data to the application, developers can make a decision on the filtering algorithm rather than narrowing down the Global Tone Detection (GTD) energy detection range set through the **dx_setgtdamp()** function at run time. This avoids invalid event filtering, and gives more control to the application to tweak tone detection filtering using its own proprietary algorithms rather than a simple low-energy threshold.

Energy detection at tone on and off events provides the necessary information to fully characterize all sorts of PSTN tones, whether single, dual, or tri-frequency tones. The application can then filter out spurious tone detections caused by noise, cross-talk and even human voice.

A new structure, FM_ENERGY, is introduced to retrieve the frame-energy-related information on Dialogic® JCT boards. This new structure is added to *dxlib.h* for DE_TONEON and DE_TONEOFF CST type events.

```

typedef struct {
    unsigned short ergy_broad;      /* Value of total energy of frame in dBm units */
    unsigned short ergy_freq1;     /* Value of main frequency peak energy, if tone being
                                   tracked in frame in dBm units */
    unsigned short ergy_freq2;     /* Value of 2nd frequency peak energy, if tone being
                                   tracked in frame in dBm units */
    unsigned short ergy_rfu;       /* Reserved for future use */
} FM_ENERGY;

```

Note: This feature is only supported on TDX_CST asynchronous event notification. Retrieval of this data structure through the **dx_getevt()** function is not available.

1.4.2 Asynchronous Event Data for CST GTD Tone Event Detection

Structure and field	Type	Value	
sr_getevttype()	long	TDX_CST	
sr_getevtdata()	DX_CST * TN_INFO * TN_TIMESTAMP * FM_ENERGY *	Pointer (address) of a DX_CST structure Pointer (address) of a TN_INFO structure Pointer (address) of a TN_TIMESTAMP structure Pointer (address) of a FM_ENERGY structure	
sr_getevtlen()	unsigned long	sizeof(DX_CST) +sizeof(TNINFO) +sizeof(TN_TIMESTAMP) +sizeof(FM_ENERGY)	
Structure DX_CST Fields			
cst_event	unsigned short	DE_TONEON	DE_TONEOFF
cst_data	unsigned short	Tone ID	Tone ID
Structure TN_INFO Fields			
tn_freq1	unsigned short	Lowest frequency component in Hz	Zero (0)
tn_freq2	unsigned short	Highest frequency component in Hz or Zero (0)	Zero (0)
tn_on	unsigned short	On time in msec or Zero (0)	Zero (0)
tn_off	unsigned short	Off time in msec or Zero (0)	Zero (0)
tn_rep_cnt	unsigned short	Repetition count or Zero (0)	Zero (0)
rfu	unsigned short	Zero (0)	Zero (0)
Structure TN_TIMESTAMP Fields			
tn_TimeStamp	unsigned long	On-board time stamp	On-board time stamp
Structure FM_ENERGY Fields			
ergy_broad;	unsigned short	Total energy on channel input	Total energy on channel input
ergy_freq1;	unsigned short	Lowest frequency component energy	Zero (0)
ergy_freq2;	unsigned short	Highest frequency component energy or Zero (0)	Zero (0)
ergy_rfu;	unsigned short	Zero (0)	Zero (0)

1.4.3 Asynchronous Event Data for CST Silence Event Detection

Structure and field	Type	Value	
sr_getevtttype()	unsigned long	TDX_CST	
sr_getevtdata()	DX_CST * TN_TIMESTAMP *	Pointer (address) of a DX_CST structure Pointer (address) of a TN_TIMESTAMP structure	
sr_getevtlen()	unsigned long	Sizeof(DX_CST) +sizeof(TN_TIMESTAMP)	
Structure DX_CST Fields			
cst_event	unsigned short	DE_SILON	DE_SILOFF
cst_data	unsigned short	Time since previous silence started in 10 msec units	Time since previous silence stopped in 10 msec units
Structure TN_TIMESTAMP Fields			
tn_TimeStamp	unsigned long	On-board time stamp	On-board time stamp

1.4.4 Example

This example demonstrates what transpires after receiving a TDX_CST event asynchronously.

```

DX_CST *datap;
TN_INFO *tonep;
TN_TIMESTAMP *tsp;
FM_ENERGY * energy;

long timestamp=0; // time stamp in ms units
unsigned short energy_broad =0, energy_freq1=0, energy_freq2=0;
switch(sr_getevtttype(ehandle))
{
case TDX_CST:
    datap = (DX_CST *) sr_getevtdata(ehandle);
    if (datap->cst_event == DE_TONEON || datap->cst_event == DE_TONEOFF)
    {

        tonep = (TN_INFO*)(datap+1); // tone structure starts at end of CST structure
        tsp = (TN_TIMESTAMP*)(tonep+1); // time stamp structure starts at end of
                                         TN_INFO structure.

        energy = (FM_ENERGY*)(tsp+1);
        timestamp = tsp->tn_TimeStamp; // get the time stamp
        energy_broad = energy->ergy_broad; // total energy on channel input
        if (datap->cst_event == DE_TONEON)
        {
            energy_freq1 = energy->ergy_freq1; //lowest frequency component energy
            energy_freq2 = energy->ergy_freq2; //highest frequency component energy
        }

    }
    if (datap->cst_event == SILON || datap->cst_event == SILOFF)
    {

        tsp = (TN_TIMESTAMP*)( datap +1); // time stamp structure starts at end of

```

```

                                TN_INFO structure.
timestamp = tsp->tn_TimeStamp; // get the time stamp

}
break;

```

1.4.5 Documentation

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

For more information, refer to the *Voice API Library Reference* and the *Voice API Programming Guide*.

1.5 Support for SFTP in Dialogic® Global Call SS7 Call Control Library

With the Service Update, a parameter has been added to the *gcss7.cfg* file to specify the type of file transfer protocol used by the Dialogic® Global Call SS7 server to retrieve configuration files from the Signal Interface Units (SIUs) when boards are downloaded. By default, the Global Call SS7 server uses regular ftp. This new parameter, **SIU.FTP_Type**, allows ssh ftp (sftp) to be used. For further information about the **SIU.FTP_Type** parameter, see the *Dialogic® Global Call SS7 Technology Guide*.

- Notes:**
1. For sftp functionality on Linux systems, make sure the OpenSSH and OpenSSL software is installed and that lftp and sftp are functional. If the OpenSSH software does not come with your Linux operating system, you can download it from <http://www.openssh.com/>. If the OpenSSL software does not come with your Linux operating system, you can download it from <http://www.openssl.org/>.
 2. If you are using Red Hat Linux Version 3.0 or earlier, you must upgrade the lftp software for sftp support. See the *Dialogic® Global Call SS7 Technology Guide* for instructions.

1.6 Raw Data Mode Support with Dialogic® DM3 Boards

With the Service Update, Dialogic® DM/V-A and DM/V-B Media Boards support recording and streaming of raw data, that is, without any transcoding, using the Dialogic® Continuous Speech Processing (CSP) API.

1.6.1 Feature Description

The following CSP API functions can be used to record and stream raw data:

- **ec_reciottdata()** to record raw data to a file
- **ec_stream()** to stream raw data

These functions point to a DX_XPB data structure for the data format. A new value for the wDataFormat field, **DATA_FORMAT_RAW**, is used for raw data mode.

The fields of the DX_XPB data structure must be set as follows for raw data mode:

wFileFormat

Set to FILE_FORMAT_VOX. (Do not use FILE_FORMAT_WAV.)

wDataFormat

Set to DATA_FORMAT_RAW for raw data mode.

nSamplesPerSec

Set to DRT_8KHZ.

wBitsPerSample

Set to 8 bits per sample.

The following guidelines apply when using raw data mode:

- Raw data mode can be used with any media load that supports CSP. All characteristics of the media load (features, densities) will apply.
- When using raw data mode, the following CSP features are **disabled**, even if specifically enabled via software:
 - Echo canceller
 - Voice activity detector (VAD)
 - Pre-speech buffer
 - Barge-in and voice event signaling
 - Silence compressed streaming
- The nSamplesPerSec and wBitsPerSample settings in DX_XPB denote the characteristics of data stream encapsulation from its end point of origin onto a synchronous 64 kbps TDM time slot, through the CT Bus, and finally onto the local end point or file.
- The format and characteristics of the output stream or file will resemble the input source and are not determined in any way by the DX_XPB settings.

Sample Code

The following is an example of **ec_stream()** using raw mode.

```
#include <stdio.h>
#include <fcntl.h>
#include <srllib.h>
#include <dxlib.h>
#include <eclib.h>

int stream_cb(int chDev, char *buffer, unsigned int length);

int main()
{
    int ret;
    int csp_dev;
    long term;
    int parmval;
    DV_TPT tpt[2];
```

```

int srlmode;
static DX_XPB xpb = {
    FILE_FORMAT_VOX,
    DATA_FORMAT_RAW,
    DRT_8KHZ,
    8
};

/* Set SRL Mode to Run in Polled Mode */
srlmode = SR_POLLMODE;
if (sr_setparm(SRL_DEVICE, SR_MODEID, (void *)&srlmode) == -1) {
    /* Process error */
}

/* Open the channel device */
if ((csp_dev = dx_open("dxxxBlCl", (int)NULL) == -1) {
    /* Process error */
}

/* Set the transfer buffer size to be 1K */
parmval = 1024;
ret = ec_setparm(csp_dev, ECCH_XFERBUFFERSIZE, (void *)&parmval);
if (ret == -1) {
    printf("Error in ec_setparm(). Err Msg = %s, Lasterror = %ld\n",
        ATDV_ERRMSGP(csp_dev), ATDV_LASTERR(csp_dev));
}

/* set up DV_TPTs for the required terminating conditions */
tpt[0].tp_type = IO_CONT;
tpt[1].tp_type = IO_EOT; /* last entry in the table */
dx_clrtp(tpt, 2);

tpt[0].tp_termno = DX_MAXDTMF; /* Maximum digits */
tpt[0].tp_length = 1; /* terminate on the first digit */
tpt[0].tp_flags = TF_MAXDTMF; /* Use the default flags */
tpt[1].tp_termno = DX_MAXTIME; /* Maximum time */
tpt[1].tp_length = 100; /* terminate after 10 secs */
tpt[1].tp_flags = TF_MAXTIME; /* Use the default flags */

/* Now stream to a memory buffer using ec_stream() */
ret = ec_stream(csp_dev, tpt, &xpb, stream_cb, EV_ASYNC | MD_NOGAIN);
if (ret == -1)
{
    printf("Error in ec_stream(). Err Msg = %s, Lasterror = %ld\n",
        ATDV_ERRMSGP(csp_dev), ATDV_LASTERR(csp_dev));
}

/* Wait for ec_stream termination event */
while (1) {
    sr_waitvt(-1);
    ret = sr_getevtttype();

    if (ret == TEC_STREAM) {
        printf("TEC_STREAM - termination event received.\n");
        break;
    }
    else if (ret == TDX_ERROR) {
        printf("ERROR event received.\n");
    }
    else {
        printf("Event 0x%x received.\n", ret);
    }
} /* end while */

```

```

        /* Retrieve the reason for the ec_stream termination */
        if ((term = ATEC_TERMMSK(csp_dev)) == AT_FAILURE) {
            /* Process error */
        }

        /* Examine bitmap to determine if digits caused termination */
        if (term & TM_MAXDTMF) {
            printf("Terminated on digits\n");
        }

        /* Close the voice channel */
        dx_close(csp_dev);
        return(0);
    }

int stream_cb(int chDev, char *buffer, unsigned int length)
{
    /* process data here ... */
    return((int)length);
}

```

1.6.2 Documentation

The online bookshelf provided with Dialogic® System Release 6.1 for Linux contains information about all system release features including features for application development, configuration, administration, and diagnostics.

For more information about the Dialogic® Continuous Speech Processing (CSP) API, see the following documents:

- *Dialogic® Continuous Speech Processing API Programming Guide*
- *Dialogic® Continuous Speech Processing API Library Reference*

For more information about the DX_XPB data structure, see the *Dialogic® Voice API Library Reference*.

1.7 File Management Enhancements for DebugAngel Tool

With the Service Update, the Dialogic® DebugAngel tool has additional command line options to provide capabilities for managing multiple log files.

1.7.1 Feature Description

The DebugAngel tool provides low-level firmware tracing, to aid in the troubleshooting of firmware issues on Dialogic® DM3 Boards. The tool is described in the *Dialogic® System Software Diagnostics Guide*. This feature enhances the file management capabilities for the log files created by DebugAngel. (Content of the log files remains unchanged.)

By default, on a Linux system, information is logged to the file *debugangel.log* in the *\$(INTEL_DIALOGIC_DIR)/log* directory. Previously, only one log file could be created and there was no maximum file size. With the new command line options, multiple log files can

be created, with several options for naming them, and the maximum log file size can be specified.

Command Line Options

In addition to options for starting/stopping the debugangel daemon, there are several new command line options. Usage is:

```
debugangel <options> [start | stop]
```

If only mandatory command line options are entered, for example:

```
debugangel start
```

then by default DebugAngel checks for an existing *debugangel.log* file and renames it. The backup file naming format is *debugangel.log.1*, *debugangel.log.2*, etc. Multiple backup files are kept.

If any command line option(s) is entered along with a start command, the tool still checks for an existing file with the same name and backs it up; however, the backup file naming format will now be the original name of the log file followed by a .bak extension. If a backup file with the same name already exists, it is overwritten. Therefore, only one .bak file is kept, for example, *debugangel.log.bak*. Furthermore, it is now possible to fully control the backup process; see the below new option “-a<#>” for more information.

Options:

start | stop (mandatory)

starts or stops the debugangel daemon.

-f<logfile name> (optional)

Specifies the file name of the log file to be used by DebugAngel. Default is *\$/[INTEL_DIALOGIC_DIR]/log/debugangel.log*, for example, */usr/dialogic/log/debugangel.log*.

If only a file name is entered, the file will be created in the default directory, *\$/[INTEL_DIALOGIC_DIR]/log*. You can also enter a relative or full path name along with the name of the file, and the file will be created in the specified directory.

Note: If a relative or full path name is used for the -f option, the target directory must exist before running the debugangel daemon.

Examples:

- `debugangel -ftestlog start`
The resulting file in the */usr/dialogic/log* directory: *testlog.log*
- `debugangel -f./test/testlog start` (running the debugangel command line from the */usr/dialogic/log* directory)
The resulting file in the */usr/dialogic/log/test* directory: *testlog.log*
- `debugangel -f/usr/dialogic/log/test123/testlog start`
The resulting file in the */usr/dialogic/log/test123* directory: *testlog.log*

-m<#> (optional)

Specifies the maximum log file size in bytes. Default is 0 (unlimited size).

When a log file reaches the specified maximum size, the logging behavior depends on the **-n** and **-a** options as explained below.

-n<#> (optional)

Specifies the maximum number of log files that should be created by DebugAngel. Default is 1 file. This setting is used in conjunction with the **-m** option. When a log file reaches the maximum log file size specified by **-m**, it is closed and a new log file is created. Multiple log files are named according to the **-a** option.

Notes: 1. If **-n** is set greater than 1, then **-m** cannot be equal to 0; it is not permitted to have multiple files of unlimited size. If you specify those settings, an error message is generated and the debugangel daemon will not start.

2. The **-n** option cannot be set less than or equal to 0. If it is, an error message is generated and the debugangel daemon will not start.

-a<#> (optional)

Auto Rename mode; controls whether an existing log file is backed up (e.g., when the start option is used), and specifies the naming convention to use when creating log files. Further information is given in the following sections:

- [Auto Rename Options for Single Log Files](#)
- [Auto Rename Options for Multiple Log Files](#)

-? (optional)

Displays the help menu for debugangel.

If any errors are encountered during processing of the command line options, an appropriate error message is output to STDOUT and DebugAngel is terminated without starting the debugangel daemon.

Auto Rename Options for Single Log Files

This section explains how the **-a** (Auto Rename) options work when there is a single log file (**-n1**).

Note: With **-n1**, the value of the **-a** option is automatically set to 1.

-a0

The log file name is the default or the value specified with the **-f** option. There is no backup of an existing log file. If the file exists when DebugAngel starts, it is deleted and replaced with a new file of the same name. With **-m0**, the file is allowed to grow without limit. With **-m > 0**, the file is allowed to grow to the specified limit. When the limit is reached, the file is truncated and logging is resumed from the beginning of the file.

-a1

The log file name is the default or the value specified with the **-f** option. If the file exists when DebugAngel starts, it is backed up and renamed with a .bak extension, to avoid overwriting the original. For example, *debugangel.log* is renamed *debugangel.log.bak*. With **-m0**, the file is allowed to grow without limit. With **-m > 0**,

the file is allowed to grow to the specified limit. When the limit is reached, the file is truncated and logging is resumed from the beginning of the file.

-a2

Adds a timestamp with the current date and time to the log file name. A file name with a timestamp has the following format:

```
filename.MM_DD_at_hh_mm_ss.zzz.log
```

where:

- filename - the default or the value specified with the **-f** option, stripped of the “.log” extension. (The “.log” extension is appended to the modified file name.)
- MM - month (01=January, 02=February, 03=March, ... 12=December)
- DD - day of the month (01-31)
- hh - hour (24-hour format, 00-23)
- mm - minute (00-59)
- ss - second (00-59)
- zzz - millisecond (000-999)

For example, if DebugAngel is started on February 17 at 3:11:27:357 p.m., with **-fDebugAngel.log**, the name of the log file created is:

```
DebugAngel.02_17_at_15_11_27.357.log
```

There is no backup of an existing log file at startup. With **-m0**, the file is allowed to grow without limit. With **-m > 0**, the file is allowed to grow to the specified limit. When the limit is reached, the file is deleted, a new log file is created (using the same naming convention), and logging is resumed. This process is repeated until logging is stopped.

-a3

Adds “00” to the log file name, before the “.log” extension. (Although this option can be used with a single log file, it is more suitable when using multiple log files, where it appends an index number to each log file name.) For example, with **-fDebugAngel.log**, the name of the log file created is:

```
DebugAngel00.log
```

There is no backup of an existing log file at startup. If the file exists when DebugAngel starts, it is deleted and replaced with a new file of the same name. With **-m0**, the file is allowed to grow without limit. With **-m > 0**, the file is allowed to grow to the specified limit. When the limit is reached, the file is deleted, a new log file is created with the same name, and logging is resumed. This process is repeated until logging is stopped.

Auto Rename Options for Multiple Log Files

This section explains how the **-a** (Auto Rename) options work when there are multiple log files (**-n > 1**).

Note: When **-n** is set **greater than 1**, the **-a** option must be set to either **2** or **3**. If **-n > 1** and **-a** is set to either **0** or **1**, an error message is generated and the debugangel daemon will not start.

-a2

Adds a timestamp with the current date and time to the log file name. A file name with a timestamp has the following format:

`filename.MM_DD_at_hh_mm_ss.zzz.log`

where:

- filename - the default or the value specified with the **-f** option, stripped of the “.log” extension. (The “.log” extension is appended to the modified file name.)
- MM - month (01=January, 02=February, 03=March, ... 12=December)
- DD - day of the month (01-31)
- hh - hour (24-hour format, 00-23)
- mm - minute (00-59)
- ss - second (00-59)
- zzz - millisecond (000-999)

For example, if DebugAngel is started on February 17 at 3:11:27:357 p.m., with **-fDebugAngel.log**, the name of the *first* log file created is:

`DebugAngel.02_17_at_15_11_27.357.log`

There is no backup of an existing log file at startup. When the file reaches its maximum size (specified with **-m**), it is closed and a new log file is created. The new log file will have a timestamp with the current date and time in its file name. This process is repeated until the maximum number of log files (specified with **-n**) is reached. When the next log file after that is created, the oldest log file is deleted so that no more than the maximum number of log files are saved at any time. See [Examples of Multiple Log Files](#) below. This process is repeated until logging is stopped.

-a3

Adds a numeric index (counter) to the log file name. The file name has the following format:

`filenamenn.log`

where:

- filename - the default or the value specified with the **-f** option, stripped of the “.log” extension. (The “.log” extension is appended to the modified file name.)
- nn - a number starting with 00, then incrementing to 01, 02, etc., up to MaxFiles-1

For example, with **-fDebugAngel.log**, the names of the log files are *DebugAngel00.log*, *DebugAngel01.log*, *DebugAngel02.log*, etc.

There is no backup of an existing log file at startup. When the file reaches its maximum size (specified with **-m**), it is closed and a new log file is created. The new log file will have the next sequential number in its file name. This process is repeated until the maximum number of log files (specified with **-n**) is reached. When the next log file after that is created, the oldest log file is deleted so that no more than the maximum number of log files are saved at any time. The file naming is repeated starting with the number 00 again. See [Examples of Multiple Log Files](#) below. This process is repeated until logging is stopped.

Examples of Multiple Log Files

With the following settings (and default log file name):

- **-a2**
- **-m65536**
- **-n5**

The resulting files in `/usr/dialogic/log` are:

```
65,536  debugangel.09_26_at_16_29_08.031.log
65,536  debugangel.09_26_at_16_33_18.000.log
65,536  debugangel.09_26_at_16_44_09.008.log
65,536  debugangel.09_26_at_16_47_12.035.log
10,871  debugangel.09_26_at_16_56_58.041.log
```

Note: When this file is filled up, the first file is removed. No more than 5 files exist at any time. Each new file created has a timestamp.

With the following settings (and default log file name):

- **-a3**
- **-m1048576**
- **-n4**

The resulting files in `/usr/dialogic/log` are:

```
1,048,576  debugangel00.log
1,048,576  debugangel01.log
1,048,576  debugangel02.log
650,355    debugangel03.log
```

Note: When this file is filled up, the first file (*debugangel00.log*) is overwritten. No more than 4 files exist at any time. The files are always named *debugangel00.log*, *debugangel01.log*, *debugangel02.log*, and *debugangel03.log*.

1.7.2 Documentation

The online bookshelf provided with Dialogic® System Release 6.1 for Linux contains information about all system release features including features for application development, configuration, administration, and diagnostics.

For more information about DebugAngel, see the *Dialogic® System Software Diagnostics Guide*.

1.8 New Media Loads for Dialogic® DM/V2400A-cPCI and DMV4800BC Media Boards

The Service Update provides a new media load, 9F, for the Dialogic® DM/V2400A-cPCI and DMV4800BC Media Boards. This new media load supports rich conferencing

(conferencing, echo cancellation, and tone clamping) and fax. Media load 9F-MC, for the DMV4800BC Board only, provides a similar set of features and also allows a larger conference size (with fewer conferences).

1.8.1 Feature Description

Predefined sets of features for Dialogic® DM3 Boards are provided in media loads. A media load consists of a configuration file set (PCD, FCD, and CONFIG files) and the associated firmware that is downloaded to the board. See the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* for more information about media loads.

The new media loads are described below (the channel densities of the features in media load 9F are different on the DM/V2400A-cPCI and DMV4800BC Media Boards, so they are discussed separately):

- [Media Load 9F for DM/V2400A-cPCI Media Board](#)
- [Media Load 9F for DMV4800BC Media Board](#)
- [Media Load 9F-MC for DMV4800BC Media Board](#)

Media Load 9F for DM/V2400A-cPCI Media Board

The features and channel densities provided by media load 9F on the DM/V2400A-cPCI Media Board are as follows:

Features Supported	Rich Conferencing with Echo Cancellation and Tone Clamping	Maximum Conference Size without Bridging	Fax
Channel Density	120	60	15

Note: Conference size is limited to 60 parties without bridging. Conference bridging can be used to effectively expand a conference beyond the maximum size. Conference bridging consumes conferencing resources, reducing overall board conference density.

Media Load 9F for DMV4800BC Media Board

The features and channel densities provided by media load 9F on the DMV4800BC Media Board are as follows:

Features Supported	Rich Conferencing with Echo Cancellation and Tone Clamping	Maximum Conference Size without Bridging	Fax
Channel Density	256	16	30

Note: Conference size is limited to 16 parties without bridging. Conference bridging can be used to effectively expand a conference beyond the maximum size. Conference bridging consumes conferencing resources, reducing overall board conference density.

Media Load 9F-MC for DMV4800BC Media Board

The features and channel densities provided by media load 9F-MC on the DMV4800BC Media Board are as follows:

Features Supported	Rich Conferencing with Echo Cancellation and Tone Clamping	Maximum Conference Size without Bridging	Fax
Channel Density	250	Mixed: One with conference size of 90, and 10 with conference size of 16 each	30

Media load 9F-MC supports a total of 11 DSP conference resources or DCB devices, each represented in the Dialogic® Audio Conferencing (DCB) API as a device `dcbBnDy`. It is the **last** DSP/DCB device with the maximum conference size of 90. Use the **`dcb_dsprescount()`** function to obtain the available conference resource count for a specified DSP.

To differentiate with the previous media load 9F, the string -MC is added to the `pcd` and configuration file names.

Note: For the smaller sized conferences, conference size is limited to 16 parties without bridging. Conference bridging can be used on the smaller sized conferences to effectively expand a conference beyond the maximum size. Conference bridging consumes conferencing resources, reducing overall board conference density.

1.8.2 Configuring the Software

The new media loads can be selected when using the Dialogic® Configuration Manager for Linux. The configuration procedure is described in detail in the *Dialogic® DM3 Architecture Products on Linux Configuration Guide*. This is done before the boards are started.

1.8.3 Documentation

The online bookshelf provided with Dialogic® System Release 6.1 for Linux contains information about all system release features including features for application development, configuration, administration, and diagnostics.

For more information about the Dialogic® Audio Conferencing (DCB) API, see the following documents:

- *Dialogic® Audio Conferencing API Programming Guide*
- *Dialogic® Audio Conferencing API Library Reference*

For detailed information about configuring Dialogic® DM/V2400A-cPCI and DMV4800BC Media Boards, see the *Dialogic® DM3 Architecture Products on Linux Configuration Guide*.

Note: The online bookshelf has not been updated for this feature, so the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* does not currently include information about media loads 9F and 9F-MC.

1.9 Support for SIP re-INVITE on Dialogic® DM/IP Boards

With the Service Update, SIP re-INVITE is now supported on Dialogic® DM/IP Boards.

For an overview of the SIP re-INVITE method and a description of the capabilities supported by the Dialogic® Global Call library, see the latest version of the *Dialogic® Global Call IP Technology Guide* that is now on the online documentation bookshelf.

Note: The Global Call SIP re-INVITE support is limited to only acknowledging the re-INVITE request by using **gc_AcceptModifyCall(NULL)**. Changing the media session parameters such as coder, direction, and remote destination address is not supported.

Note: When using SIP re-INVITE, the following parameter must be enabled in the .config file for the DM/IP Board:

```
[0x40] SetParm=0x400a,1 ! PrmEarlyMedia (0=Disabled, 1=Enabled)
```

(The configuration parameter name refers to early media because the same firmware capabilities that are required to support early media operation are also required to support acting on re-INVITE requests.)

1.10 Analog Call Transfer Support on Dialogic® Springware Boards

With the Service Update, blind and supervised analog call transfers using the Dialogic® Global Call API are now supported on Dialogic® Springware Boards. For further information, see the *Dialogic® Global Call Analog Technology Guide*.

1.11 PBX Integration Support for Nortel BCM

The Service Update adds support for the Nortel Business Communications Manager (BCM) when using the Dialogic® D/82JCT-U Board. For detailed information including programming requirements, see the *Dialogic® PBX Integration Board User's Guide*.

1.12 Configuring SIP Stack Parameters with Global Call

With the Service Update, selected SIP stack parameters such as timers can now be configured with the Dialogic® Global Call API.

1.12.1 Feature Description

A new data structure, SIP_STACK_CFG, is used to configure SIP stack parameters. Details about the SIP_STACK_CFG data structure fields are given in [Section 1.12.2, “SIP_STACK_CFG Data Structure”](#), on page 48.

To support SIP stack configuration, IP_VIRTBOARD has been updated with a new structure pointer (default is NULL) as follows:

```
typedef struct {  
    ...  
    ...  
    /* The following is added for VIRTBOARD_VERSION_SIP_STACK_CFG support */  
    SIP_STACK_CFG *sip_stack_cfg;  
    /* end VIRTBOARD_VERSION_SIP_STACK_CFG additions */  
} IP_VIRTBOARD;
```

1.12.2 SIP_STACK_CFG Data Structure

The SIP_STACK_CFG structure definition has been added in the *gcip.h* file. The new data structure is described below.

Note: SIP stack parameters can only be configured once per virtual board (at **gc_Start()**) and remain in effect throughout the Global Call application (per process).

SIP_STACK_CFG

```
typedef struct {
    unsigned long version; /* version set by INIT_SIP_STACK_CFG */
    int retransmissionT1;
    int retransmissionT2;
    int retransmissionT4;
    int generalLingerTimer;
    int inviteLingerTimer;
    int provisionalTimer;
    int cancelGeneralNoResponseTimer;
    int cancelInviteNoResponseTimer;
    int generalRequestTimeoutTimer;
} SIP_STACK_CFG;
```

■ Description

The SIP_STACK_CFG data structure is used to configure selected SIP stack parameters such as timers.

The SIP_STACK_CFG data structure is referenced by the IP_VIRTBOARD data structure, which stores configuration and capability information about an IPT (virtual) board device that is populated when the device is started. An array of IP_VIRTBOARD structures (one per virtual board in the system) is referenced by the IPCCLIB_START_DATA structure, which is passed to the **gc_Start()** function.

Applications should use the **INIT_SIP_STACK_CFG()** function to initialize the structure with the correct version number and initial field values before setting the appropriate values.

■ Field Descriptions

The fields of the SIP_STACK_CFG data structure are:

version

The version number of the data structure. The correct value is set by the **INIT_SIP_STACK_CFG()** initialization function and should not be overridden.

retransmissionT1

Determines several timers as defined in RFC 3261. For example, when an unreliable transport protocol is used, a Client Invite transaction retransmits requests at an interval that starts at T1 milliseconds and doubles after every retransmission. A Client General transaction retransmits requests at an interval that starts at T1 and doubles until it reaches T2. The default value is 1000.

retransmissionT2

Determines the maximum retransmission interval as defined in RFC 3261. For example, when an unreliable transport protocol is used, general requests are retransmitted at an interval that starts at T1 and doubles until it reaches T2. If a provisional response is received, retransmissions continue but at an interval of T2. The parameter value cannot be less than 4000. The default value is 8000.

retransmissionT4

Determines the amount of time the network takes to clear messages between client and server transactions as defined in RFC 3261. For example, when working with an unreliable transport

protocol, T4 determines the time that a UAS waits after receiving an ACK message and before terminating the transaction. The default value is 10000.

generalLingerTimer

After a server sends a final response, the server cannot be sure that the client has received the response message. The server should be able to retransmit the response upon receiving retransmissions of the request for generalLingerTimer milliseconds. The default value is 32000.

inviteLingerTimer

After sending an ACK for an INVITE final response, a client cannot be sure that the server has received the ACK message. The client should be able to retransmit the ACK upon receiving retransmissions of the final response for inviteLingerTimer milliseconds. The default value is 32000.

provisionalTimer

The provisionalTimer is set when receiving a provisional response on an Invite transaction. The transaction will stop retransmissions of the Invite request and will wait for a final response until the provisionalTimer expires. If you set the provisionalTimer to 0, no timer is set, and the Invite transaction will wait indefinitely for the final response. The default value is 180000.

cancelGeneralNoResponseTimer

When sending a CANCEL request on a General transaction, the User Agent waits cancelGeneralNoResponseTimer milliseconds before timeout termination if there is no response for the canceled transaction. The default value is 32000.

cancelInviteNoResponseTimer

When sending a CANCEL request on an Invite request, the User Agent waits cancelInviteNoResponseTimer milliseconds before timeout termination if there is no response for the canceled transaction. The default value is 32000.

generalRequestTimeoutTimer

After sending a General request, the User Agent waits for a final response generalRequestTimeoutTimer milliseconds before timeout termination (in this time the User Agent retransmits the request every T1, 2*T1, ... , T2, ... milliseconds). The default value is 32000.

1.12.3 Sample Code

The following example sets the SIP T1 timer to 64 ms.

```
#include "gclib.h"
..
..
#define BOARDS_NUM 1
..
..

/* initialize start parameters */
IPCCLIB_START_DATA cclibStartData;
memset(&cclibStartData,0,sizeof(IPCCLIB_START_DATA));
IP_VIRTBOARD virtBoards[BOARDS_NUM];
memset(virtBoards,0,sizeof(IP_VIRTBOARD)*BOARDS_NUM);

/* initialize start data */
INIT_IPCCLIB_START_DATA(&cclibStartData, BOARDS_NUM, virtBoards);

/* initialize virtual board */
INIT_IP_VIRTBOARD(&virtBoards[0]);

/* sip stack cfg support */
SIP_STACK_CFG sip_stack_cfg;
INIT_SIP_STACK_CFG(&sip_stack_cfg);
    virtBoard[bid].sip_stack_cfg = &sip_stack_cfg;

    sip_stack_cfg.retransmissionT1 = 64;
```

1.12.4 Documentation

The online bookshelf provided with Dialogic® System Release 6.1 for Linux contains information about all system release features including features for application development, configuration, administration, and diagnostics.

For more information about the Dialogic® Global Call API in general, see the following documents:

- *Dialogic® Global Call API Programming Guide*
- *Dialogic® Global Call API Library Reference*

For features specific to IP technology, see:

- *Dialogic® Global Call IP Technology Guide*

1.13 Disabling Automatic re-INVITE Message when Switching between Fax and Audio

When using SIP, a change between audio and fax modes may cause both sides of the call to send a re-INVITE message to renegotiate the media session between them. This would cause a glare condition, which disconnects the call.

With the Service Update, the ability to disable/enable the sending of the automatic SIP re-INVITE message upon media switch can now be configured with the Dialogic® Global Call API to prevent this glare condition.

1.13.1 Feature Description

Overview of Use Case

A user application can enable and disable the unsolicited GCEV_EXTENSION notification events associated with certain types of transition events, including media streaming connection state changes. The application can receive notification of changes in the status (connection and disconnection) of media streaming in the transmit and receive directions as GC_EXTENSION_EVT events. The events for this notification must be enabled by setting or adding the bitmask value EXTENSION_EVT_SIGNALING_STATUS to the GC_EXTENSION_EVT mask. Events can be enabled on a per board basis (using **gc_SetConfigData()**) or on a per channel basis (using **gc_SetUserInfo()**).

A user application needs to enable media streaming status EXTENSION_EVT_STREAMING_STATUS to get notification of media transmit and receive connected events before doing specific media tests just after another media test is completed. This is particularly useful in back-to-back testing, because in live applications there are other indications of media session events, e.g., fax CNG/CED tones, busy tone, phone hang-up tone, etc., which are not available in back-to-back testing.

For example, consider two user applications where one makes an IP call to the other, sends a fax (over IP), and after the fax session is completed, dials a string of DTMF digits for the other side to detect. If the DTMF digits are dialed before the fax session completely ends, the DTMF dial test will fail, since the media session has not switched from fax to audio yet. In order for the application to know when to dial the DTMF digits it has to know when the previous fax session has ended and the audio session has started. It knows this when it receives an event indicating that the audio media stream is connected.

When working with the **H.323** protocol, this functionality to detect the media switch is sufficient for user applications. However for **SIP** protocols, when a fax to audio switch occurs, both sides send a re-INVITE message to renegotiate the media session between them, which causes a glare condition that drops the call. This is not an expected situation for a user application.

A similar situation can occur when the media switches from audio to fax.

New Parameters to Disable/Enable Automatic re-INVITE Messages

In order to prevent this glare situation, new parameters are now available in Global Call to:

- prevent sending an automatic SIP re-INVITE when a switch from fax to audio media occurs, or when a switch from audio to fax media occurs
- re-enable the sending of an automatic SIP re-INVITE when a switch from fax to audio media occurs, or when a switch from audio to fax media occurs

The new parameter IDs are added for the existing IPSET_CONFIG set ID as shown in the following table.

Set ID	Parameter ID	Set	Send	Retrieve	SIP/ H.323
IPSET_CONFIG	IPPARM_SIP_FAXTOAUDIO_AUTO_REINVITE_DISABLE	gc_SetConfigData() gc_SetUserInfo()	---	---	SIP only
	IPPARM_SIP_FAXTOAUDIO_AUTO_REINVITE_ENABLE	gc_SetConfigData() gc_SetUserInfo()	---	---	SIP only
	IPPARM_SIP_AUDIOTOFAAX_AUTO_REINVITE_DISABLE	gc_SetConfigData() gc_SetUserInfo()	---	---	SIP only
	IPPARM_SIP_AUDIOTOFAAX_AUTO_REINVITE_ENABLE	gc_SetConfigData() gc_SetUserInfo()	---	---	SIP only

By default, SIP re-INVITE messages upon media switch are sent automatically. The user application has to specifically disable the transmission of the re-INVITE by using the IPSET_CONFIG parameters. Typically, the automatic re-INVITE messages would be disabled on one side user application only, namely a fax server type of application that receives faxes.

The user application has to know whether to use this functionality depending on how the application is to be used. This is usually done at the start of an application. If the automatic re-INVITE messages are disabled on two applications in a back-to-back test, the switch from fax to audio will never occur, because neither side will send out a re-INVITE message to the other to renegotiate new media (audio) when a fax session ends.

The automatic re-INVITE messages can be disabled/enabled on a board, line, or call reference number (CRN) device basis. Code examples are shown below.

1.13.2 Sample Code

Disabling Transmission of Automatic re-INVITE on a Board Device

```
DisableFTToAReinvite()
{
    LINEDEV linedevbp;
    long request_id = 0;
    GC_PARM_BLK *target_datap = NULL;

    if (gc_OpenEx(&linedevbp, "N_ipTb1:P_IP", EV_SYNC, NULL) != GC_SUCCESS)
    {
        //print error
        return(FALSE);
    }

    gc_util_insert_parm_val (&target_datap, IPSET_CONFIG,
                           IPPARM_SIP_FAXTOAUDIO_AUTO_REINVITE_DISABLE,
                           sizeof (int), IP_MANUAL_MODE);
}
```

```

        if (gc_SetConfigData(GCTGT_CCLIB_NETIF, linedevbp, target_datap, 1000,
                             GCUPDATE_IMMEDIATE, &request_id, EV_ASYNC) != GC_SUCCESS)
        {
            //print error
            rcode=FALSE;
        }

        gc_util_delete_parm_blk(target_datap);

        target_datap = NULL;
    }

```

Re-Enabling Transmission of Automatic re-INVITE on a Line Device

```

EnableFTToAReinvite()
{
    GC_PARM_BLK *target_datap = NULL;

    if ((gc_util_insert_parm_val(&target_datap, IPSET_CONFIG,
                                IPPARM_SIP_FAXTOAUDIO_AUTO_REINVITE_ENABLE,
                                sizeof(int), NULL)) != GC_SUCCESS)
    {
        fprintf(stderr, "E%04d(%s): gc_SetUserInfo(line_dev = %d) failed, \n", Res[extts]-
>network.ts_ldev);
        fflush(stderr);
    }

    if ((gc_SetUserInfo(GCTGT_GCLIB_CHAN, Res[extts]->network.ts_ldev, target_datap,
                        GC_ALLCALLS)) != GC_SUCCESS)
    {
        fprintf(stderr, "E%04d(%s): gc_SetUserInfo(line_dev = %d) failed, \n", Res[extts]-
>network.ts_ldev);
        fflush(stderr);
    }
    gc_util_delete_parm_blk(target_datap);
    target_datap = NULL;
}

```

1.13.3 Documentation

The online bookshelf provided with Dialogic® System Release 6.1 for Linux contains information about all system release features including features for application development, configuration, administration, and diagnostics.

For more information about the Dialogic® Global Call API in general, see the following documents:

- *Dialogic® Global Call API Programming Guide*
- *Dialogic® Global Call API Library Reference*

For features specific to IP technology, see:

- *Dialogic® Global Call IP Technology Guide*

1.14 AMD Opteron Server Support

With the Service Update, Dialogic® System Release 6.1 for Linux has been validated for use with Advanced Micro Devices, Inc. (AMD) Opteron server processors.

1.15 IP Multicast Client Support

IP Multicast client, which was supported in previous system releases, is now supported in Dialogic® System Release 6.1 for Linux on Dialogic® DM/IP Boards.

IP Multicast is a one-to-many protocol that provides a scalable solution that efficiently uses network resources and bandwidth. Unlike IP Unicast, which requires *X* copies of data to be transmitted from a source to each of *X* number of receivers, IP Multicast allows a source to transmit only a single copy of the data being sent to multiple receivers. Routers throughout the network intelligently forward the data only to those IP endpoints that have requested it.

To enable IP Multicast client when starting a session, set the **ipm_StartMedia()** function **eDirection** parameter to **DATA_MULTICAST_CLIENT**.

Note: Alternatively, you can set the **ipm_SetRemoteMediaInfo()** function **eDirection** parameter to **DATA_MULTICAST_CLIENT** to enable IP Multicast client. However, **ipm_SetRemoteMediaInfo()** is deprecated and is included in the library for backwards compatibility only. Application developers should use the **ipm_StartMedia()** function instead of **ipm_SetRemoteMediaInfo()**.

Note: The **ipm_ModifyMedia()** function does **not** support the **DATA_MULTICAST_CLIENT** or **DATA_MULTICAST_SERVER** direction mode. The session **eDirection** setting cannot be modified once it is set.

The following limitations apply when implementing IP Multicast client:

- For dual span DM/IP Boards, the maximum supported number of simultaneous IP media channels configured for Multicast client is limited to a TDM line (24 for T1 and 30 for E1).
- IP Multicast loopback (i.e., Multicast server and Multicast client channels on the same DM/IP Board and configured for the same Multicast group) is not supported.

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*

Code Example

The following code example shows how to start a Multicast client session. There is a subtle difference in the way the API is used when starting a Multicast client session vs. starting other types of IP sessions. When starting a Multicast client session, the IP

address and coder settings of the Multicast group are specified as the **local** RTP port and **local** coder respectively. (When starting a Multicast server session, the IP address and coder settings of the Multicast group are specified as the remote RTP port and remote coder.)

```
#include <stdio.h>
#include <srllib.h>
#include <ipmlib.h>

int    nMulticastGroupPort    = 2500;
char   *szMulticastGroupAddress = "225.0.0.1";
char   *szDeviceName          = "ipmB1C1";

void StartMulticastClient(void)
{
    int nDeviceHandle;
    IPM_MEDIA_INFO MediaInfo;

    // Open an IP Media Channel
    nDeviceHandle = ipm_Open(szDeviceName, NULL, EV_SYNC);
    if (nDeviceHandle == -1) {
        printf("Failure Opening IP Media Channel %s", szDeviceName);
        // Perform Error Processing
    }

    // Join the IP Media Channel to a Multicast Group. Note that the
    // Multicast Group address is specified as the Local RTP Port Information
    MediaInfo.unCount = 2;
    MediaInfo.MediaData[0].eMediaType = MEDIATYPE_AUDIO_LOCAL_RTP_INFO;
    MediaInfo.MediaData[0].mediaInfo.PortInfo.unPortId = nMulticastGroupPort;
    strcpy(MediaInfo.MediaData[0].mediaInfo.PortInfo.cIPAddress,
           szMulticastGroupAddress);

    // NOTE: For Multicast Client processing, we do not need to specify
    //        RTCP port information

    MediaInfo.MediaData[1].eMediaType = MEDIATYPE_AUDIO_LOCAL_CODER_INFO;
    MediaInfo.MediaData[1].mediaInfo.CoderInfo.eCoderType =
        CODER_TYPE_G711ULAW64K;
    MediaInfo.MediaData[1].mediaInfo.CoderInfo.eFrameSize =
        (eIPM_CODER_FRAMESIZE)30;
    MediaInfo.MediaData[1].mediaInfo.CoderInfo.unFramesPerPkt = 1;
    MediaInfo.MediaData[1].mediaInfo.CoderInfo.eVadEnable = CODER_VAD_DISABLE;
    MediaInfo.MediaData[1].mediaInfo.CoderInfo.unCoderPayloadType = 0;
    MediaInfo.MediaData[1].mediaInfo.CoderInfo.unRedPayloadType = 0;

    // In Multicast Client mode, we are only going to receive data and will
    // not transmit any data. Therefore, we don't need to specify any Remote
    // RTP/RTCP or Coder settings.

    // Start the Multicast Client Session
    if (ipm_StartMedia(nDeviceHandle,
                      &MediaInfo,
                      DATA_MULTICAST_CLIENT,
                      EV_SYNC) == -1) {
        printf("ipm_StartMedia() failed for device=\"%s\" with error=%d\n",
              ATDV_NAMEP(nDeviceHandle), ATDV_LASTERR(nDeviceHandle));
        // Perform Error Processing
    }
}
```



```

/*
 *
 * Continue Processing
 *
 */
}

```

Note the use of eMediaType values **MEDIATYPE_AUDIO_LOCAL_RTP_INFO** and **MEDIATYPE_AUDIO_LOCAL_CODER_INFO** in the code example. They are equivalent to **MEDIATYPE_LOCAL_RTP_INFO** and **MEDIATYPE_LOCAL_CODER_INFO**, which were used in previous releases and are still supported. The following eMediaType equates are provided for backward compatibility (the generic names are equated with audio port values). Using the newer values (with “_AUDIO_”) is recommended:

```

#define MEDIATYPE_LOCAL_CODER_INFO MEDIATYPE_AUDIO_LOCAL_CODER_INFO
#define MEDIATYPE_LOCAL_RTCP_INFO MEDIATYPE_AUDIO_LOCAL_RTCP_INFO
#define MEDIATYPE_LOCAL_RTP_INFO MEDIATYPE_AUDIO_LOCAL_RTP_INFO
#define MEDIATYPE_REMOTE_CODER_INFO MEDIATYPE_AUDIO_REMOTE_CODER_INFO
#define MEDIATYPE_REMOTE_RTCP_INFO MEDIATYPE_AUDIO_REMOTE_RTCP_INFO
#define MEDIATYPE_REMOTE_RTP_INFO MEDIATYPE_AUDIO_REMOTE_RTP_INFO

```

1.16 Global DPD Enabled on Dialogic® Springware Boards

With the Service Update, Global Dial Pulse Detection (DPD) is now available by default via software. Previously, this feature had to be enabled from the factory or by ordering a separate GDPD enablement package to enable DPD on a board.

Global DPD is supported on Dialogic® Springware Boards, such as Dialogic® JCT Media Boards and Dialogic® D/4PCIU Media Boards. Global DPD is **not** supported on Dialogic® D/42JCT and D/82JCT PBX Integration Boards or on Dialogic® DM3 Media Boards.

For information about implementing Global DPD, see the *Dialogic® Voice API Programming Guide*.

1.17 Setting Data Transfer Buffer Size below 1K for Dialogic® Springware Boards

With the Service Update, it is now possible to set the data transfer buffer size (i.e., transfer rate for plays and records) to 256 or 512 bytes on Dialogic® Springware Boards. Previously, the minimum buffer size on Springware Boards was 1 Kbyte. A smaller buffer size is more efficient for applications where packet sizes are smaller than 1 Kbyte. The buffer size must be set on a **system-wide** basis (not per board or per channel).

The ability to set the buffer size to 256 or 512 bytes is supported on all Springware Voice Boards.

1.17.1 Feature Description

By default, Springware Boards use 2 x 512-byte driver-firmware buffers for plays and records, allowing the driver and firmware to transfer 512 bytes in one transaction. However, to minimize host CPU load and disk access activity, the Dialogic® Voice library feeds more than 512 bytes to/from the driver, with a default setting of 16 Kbytes. This setting can be changed via the **dx_setchxfercnt()** function. In a **high density** system, sending large data blocks to the driver decreases the number of context switches and gives the system a higher tolerance for audio gaps. However, in a **low density** system with small packet size, sending large data blocks to the driver could lead to delays.

Prior to this release, the minimum data transfer buffer size was 1 Kbyte. Setting the buffer size and transfer rate below 1K requires reducing the driver-firmware buffers below the 512-byte default. In order to do so, the following steps are required:

- **Modifying the voice.prm File** - For a 256-byte data transfer buffer size, set the firmware play and record buffer sizes to 128 bytes in the *voice.prm* file. For a 512-byte data transfer buffer size, set the firmware play and record buffer sizes to 256 bytes in the *voice.prm* file. This must be done before starting the system and takes effect on a system-wide basis.
- **Setting the Data Transfer Buffer Size with dx_setchxfercnt()** - Set the data transfer buffer size for a channel to 256 or 512 bytes using the **dx_setchxfercnt()** function before any **dx_play()** or **dx_rec()** is called. The 256- and 512-byte settings are new parameter values for the **dx_setchxfercnt()** function **bufsize_identifier** parameter. Previously, the allowable buffer sizes ranged from 1K to 32K (with 16K as the default).

Note: Using lower transfer rates increases the number of interrupts, thereby increasing the context switches and the CPU load. Therefore, this feature is recommended for low density systems.

Disk access time plays a vital role when transfer buffer sizes are lowered. For this feature to work efficiently, the system must have enough CPU time and fast access disk. If used on a system with close to 100% CPU utilization and slow I/O devices, audio gaps are likely to occur.

1.17.2 Modifying the voice.prm File

The *voice.prm* file is in the *data* subdirectory under your Dialogic® system software installation. To set the firmware play and record buffer size, edit *voice.prm* and add **PARAM 246** and **PARAM 247** as shown in the examples below:

For 256-byte data transfer buffer size:

```
#beginning of voice.prm
AREA=VOICE
SIZE=WORD
BASE=DECIMAL
. . .
PARAM 246 : 128 # set firmware play buffer size to 128 bytes
PARAM 247 : 128 # set firmware record buffer size to 128 bytes
#end of voice.prm
```

For 512-byte data transfer buffer size:

```
#beginning of voice.prm
AREA=VOICE
SIZE=WORD
BASE=DECIMAL
. . .
PARAM 246 : 256 # set firmware play buffer size to 256 bytes
PARAM 247 : 256 # set firmware record buffer size to 256 bytes
#end of voice.prm
```

The *voice.prm* file is read by the firmware at download time to set the buffer size to the right value.

1.17.3 Setting the Data Transfer Buffer Size with `dx_setchxfercnt()`

The `dx_setchxfercnt()` function sets the data transfer buffer size for a channel. This function can change the size of the buffer used to transfer voice data between a user application and the board. Two new values for the **bufsize_identifier** parameter are now available to set the buffer size to 256 or 512 bytes. Values for buffer size can range from 256 bytes to 32 Kbytes and are specified as follows:

bufsize_identifier - specifies the bulk queue buffer size for the channel. Use one of the following values:

- 0 - sets the buffer size to 4 Kbytes
- 1 - sets the buffer size to 8 Kbytes
- 2 - sets the buffer size to 16 Kbytes (default)
- 3 - sets the buffer size to 32 Kbytes
- 4 - sets the buffer size to 2 Kbytes
- 5 - sets the buffer size to 1 Kbytes
- 6 - sets the buffer size to 1.5 Kbytes
- 7 - sets the buffer size to 256 bytes
- 8 - sets the buffer size to 512 bytes

For example, to set the data transfer buffer size for a channel to 512 bytes, call `dx_setchxfercnt()` before any `dx_play()` or `dx_rec()`, as follows:

```
dx_setchxfercnt(chdev, 8)
```

1.17.4 Documentation

The online bookshelf provided with Dialogic® System Release 6.1 for Linux contains information about all system release features including features for application development, configuration, administration, and diagnostics.

For more information about configuring Dialogic® Springware Boards, including information about the *voice.prm* file, see the *Dialogic® Springware Architecture Products on Linux Configuration Guide*.

For more information about the Dialogic® Voice API, see the following documents:

- *Dialogic® Voice API Programming Guide*
- *Dialogic® Voice API Library Reference*

Note: The online bookshelf has not been updated for this feature, so the manuals above do not contain information about setting the data transfer buffer size below 1K.

1.18 New Media Load for Dialogic® DMV1200BTEP Media Boards

The Service Update provides a new media load, 10b, for the Dialogic® DMV1200BTEP Media Board. This new media load provides rich conferencing (conferencing, echo cancellation, and tone clamping) while also providing full density basic voice with transaction record and FSK.

1.18.1 Feature Description

Predefined sets of features for Dialogic® DM3 Boards are provided in media loads. A media load consists of a configuration file set (PCD, FCD, and CONFIG files) and the associated firmware that is downloaded to the board. See the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* for more information about media loads.

The features and channel densities provided by media load 10b are as follows:

Features Supported	Basic Voice	Transaction Record	FSK	Conferencing with Echo Cancellation and Tone Clamping
Channel Density	120	120	120	120

There are 120 total voice resources. Any combination of the voice features (basic voice, transaction record, and FSK) can be used up to a total of 120. In addition to these voice resources, 120 conferencing resources (with echo cancellation and tone clamping) can be used.

Note: Conference size is limited to 20 parties without bridging. Conference bridging can be used to effectively expand a conference beyond the maximum size. Conference bridging consumes conferencing resources, reducing overall board conference density.

Media load 10b can be used with all protocols supported on the Dialogic® DMV1200BTEP Board, e.g., T1 ISDN, T1 CAS, E1 ISDN, E1 R2MF, and DPNSS/DASS2.

1.18.2 Configuring the Software

The new media load can be selected when using the Dialogic® Configuration Manager for Linux. The configuration procedure is described in detail in the *Dialogic® DM3 Architecture Products on Linux Configuration Guide*. This is done before the boards are started.

1.18.3 Documentation

The online bookshelf provided with Dialogic® System Release 6.1 for Linux contains information about all system release features including features for application development, configuration, administration, and diagnostics.

For detailed information about configuring Dialogic® DMV1200BTEP Boards, see the *Dialogic® DM3 Architecture Products on Linux Configuration Guide*.

Note: The online bookshelf has not been updated for this feature, so the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* does not currently include information about media load 10b.

1.19 Troubleshooting Information for RTF Logs

To assist in troubleshooting, a table showing runtime and firmware errors that may appear in Dialogic® Runtime Trace Facility (RTF) logs is now available. You can get a description of errors and the suggested action to resolve the error. To access the table, use this link:

- [Error Code Table](#)

For runtime errors, the table provides the following information:

Internal error value

The error code detected internally by the library. In some of the libraries, more than one internal error is mapped to an end user error. When contacting support about failures, this information will be helpful to the support engineer because it provides more specific information about why the error was generated. This number may appear in the RTF log (with the end user error value).

Note: Sometimes the internal error value and end user error value are listed in the same trace entry. Sometimes the internal error value may appear as a separate entry.

End user error

The name of the constant that is documented in the library API reference.

End user error value

The numeric value of the constant that is documented in the library API reference. This is the value that will appear in the RTF log, which you can then search for in the table.

Description of the error

A textual description of the error.

Action to be taken

The suggested action to resolve the error.

For firmware errors, the table provides the following information:

Resource

The firmware entity in which the error occurred. A resource is technically called a DM3 resource and is a software entity that provides a service to other DM3 resources. You can use the resource information to better narrow down what activity was occurring when the error occurred.

Loc hex

The value that will appear in the RTF log (for example, 0x80000C), which you can then search for in the table.

Error class

A classification of the firmware error in broad categories. You can use this column to understand the type of action to take for a particular type of error. For example, if an error is classified as a memory error, action can be taken that is specific to this type of error (such as a pool configuration change).

Error subclass

Provides a bit more specialization with regard to the error class. Whenever possible, if a class could be subdivided into more specific classifications, it was done. The use of the error subclass is the same as that of the error class.

Action to be taken

The suggested action to resolve the error.

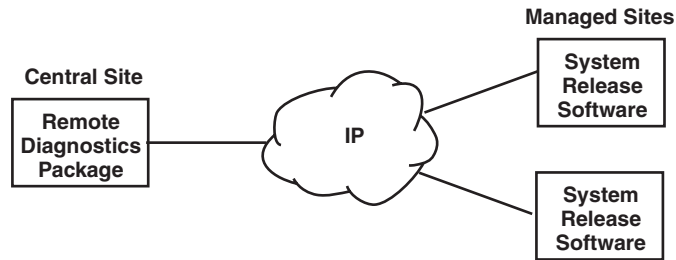
1.20 Support for LiS 2.19.1

With the Service Update, Linux STREAMS (LiS) version 2.19.1 is now provided with the Dialogic® system release software (instead of version 2.18.1). LiS version 2.19.1 is backward compatible and can be used with all supported operating systems.

1.21 Remote Diagnostics Package

A remote diagnostics package is now available that allows you to run Dialogic® diagnostics utilities remotely from a central site. The managed sites must have Dialogic®

System Release 6.1 for Linux installed. The central site does **not** need Dialogic® System Release 6.1 for Linux installed.



The remote diagnostics package is a *subset* of the system release software. It is designed for managing multiple remote sites from a central site, where the central site does not need the system software release or any Dialogic® boards installed. Instead, the remote diagnostics package must be installed at the central site. The diagnostics utilities in the remote diagnostics package are the same as the diagnostics utilities in Dialogic® System Release 6.1 for Linux.

1.21.1 Diagnostics Utilities

The remote diagnostics package includes the following utilities:

- Diagnostics Management Console (DMC)
- Runtime Trace Facility Manager and Server application (RTFManager, RTFServer)

For information about these utilities, see the *Dialogic® System Software Diagnostics Guide* on the online bookshelf for Dialogic® System Release 6.1 for Linux.

1.21.2 Installing the Remote Diagnostics Package

The remote diagnostics package can be downloaded from the Dialogic support website.

Requirements at central site:

- SSH client
- IP connectivity to managed sites
- Java Runtime Environment (JRE) version 1.5 or later

Requirements at managed sites:

- SSH server
- IP connectivity to central site
- Dialogic® System Release 6.1 for Linux installed

After installing the remote diagnostics package, you must activate the Dialogic environment variables before any of the diagnostics utilities can be run. Choose from one of the following methods:

- Run 'source /etc/profile.d/dlgc_rmtdiag.sh'.
- Log out and log back in.
- Reboot the system.

If you intend to use the RTF Server in remote logging mode, you must start the server after modifying the *RtfConfigLinux.xml* file. To start the RTF Server, choose from one of the following methods:

- Run '/etc/init.d/ dlgc_rmtdiag start'.
- Reboot the system.

1.22 Enhanced Diagnostics Tools

The Service Update introduces enhanced versions of the following diagnostics tools:

- [PSTN Diagnostics \(pstndiag\)](#)
- [Status Monitor \(statusmon\)](#)

Notes: 1. Java Runtime Environment (JRE) version 1.5 or later must be installed on your system to run the new diagnostics tools. However, there is also a console version of the PSTN Diagnostics tool that does not require JRE to be installed. See [Console Version of PSTN Diagnostics \(pstndiagc\)](#) for information about this version of the tool.

2. These tools are used with Dialogic® DM3 Boards only.

1.22.1 PSTN Diagnostics (pstndiag)

The PSTN Diagnostics tool (pstndiag) is a utility for diagnosing and troubleshooting call control issues on public switched telephone network (PSTN) connections.

The pstndiag tool has a graphical user interface (GUI). When you start the tool, a tree view of all installed Dialogic® DM3 Boards is displayed. The view can be expanded to show the lines (trunks) on each board and the channels on each line. At each level (board, line, channel), different diagnostics activities can be launched, for example:

- At the board level, you can display board configuration (board name, board number, number of lines, number of channels per line, and signaling type). You can also launch the statusmon tool. (The new statusmon tool is described in [Section 1.22.2, "Status Monitor \(statusmon\)"](#), on page 66.)
- At the line level, you can launch the lineadmin tool to put lines in/out of service, generate transmit alarms, enable/disable various types of loopbacks, and report bipolar violations, consecutively errored seconds, frame errors, and other saturation alarms.

- At the channel level, you can launch the phone tool to perform call control operations. You can also trace all call related activity on a given channel and store it in a columnar format based on timestamp deltas.

Running the PSTN Diagnostics Tool

To run the **new** version of pstndiag, enter the command:

- `pstndiag -j`

(The previous version of the tool is still supported and can be run by entering the command `pstndiag` without the `-j`.)

The new version of pstndiag includes the following changes:

- Faster startup
- Changes in the board tree view
- Additional features in the lineadmin tool: enabling all supported loopback modes and counters for saturation alarms
- Configurable modes of operation for the phone tool: basic, advanced, and expert

Note: More detailed information about the new version of pstndiag will be provided in the *Dialogic® System Software Diagnostics Guide*, which is scheduled to be updated soon.

Console Version of PSTN Diagnostics (pstndiagc)

The console version of the PSTN Diagnostics tool (pstndiagc) is designed for customers who configure Linux systems without a graphical desktop environment or for systems that are headless (no monitor). These systems can be managed remotely via a telnet or SSH session that provides a simple terminal interface. The pstndiagc tool doesn't provide the full functionality of pstndiag (due to its lack of a GUI), but it provides basic call control, line control, and alarm monitoring.

To run pstndiagc, enter the command:

- `pstndiagc`

Optional command line arguments are:

- v
show version information
- h
show location of the help file. The help file describes the layout of the pstndiagc screen and lists the commands that can be entered (such as making calls, answering calls, etc.).

1.22.2 Status Monitor (statusmon)

The Status Monitor tool (statusmon) is a utility for monitoring the current activity on all lines and channels on a Dialogic® DM3 Board. The primary use case is as a long-term monitoring tool.

The statusmon tool displays the following information:

- Alarm status (red, yellow, LOS)
- Channel state
- Call state

Running the Status Monitor Tool

The statusmon tool is typically launched from pstndiag, but it can also be run on its own. To run the **new** version of statusmon, enter the command:

- `run_statusmon.sh -board #`

where # is the logical board number of the board to monitor.

(The previous version of the tool is still supported and can be run by entering the command `statusmon board` or `statusmon board trunk channel`.)

The new version of statusmon includes the following changes:

- No line (trunk) or channel mode. However, these capabilities are supported via the pstndiag tool.

Note: More detailed information about the new version of statusmon will be provided in the *Dialogic® System Software Diagnostics Guide*, which is scheduled to be updated soon.

1.23 Support for PCI Express Boards - Dialogic® Station Interface Boards

With the Service Update, Dialogic® System Release 6.1 for Linux now supports the following Dialogic® Station Interface Boards in the PCI Express form factor:

Dialogic® DISI16-EW Switching Board

Provides connectivity for up to 16 station interfaces in a single, full-length PCI Express slot. Includes conferencing, voice play/record, tone detection and generation, and caller ID capabilities.

Dialogic® DISI24-EW Switching Board

Provides connectivity for up to 24 station interfaces in a single, full-length PCI Express slot. Includes conferencing, voice play/record, tone detection and generation, and caller ID capabilities.

Dialogic® DISI32-EW Switching Board

Provides connectivity for up to 32 station interfaces in a single, full-length PCI Express slot. Includes conferencing, voice play/record, tone detection and generation, and caller ID capabilities.

When configuring the system for the PCI Express form factor boards, use the same menu selections and configuration settings that are documented for the PCI version of the boards.

Note: When installing the Dialogic® DISI16-EW, DISI24-EW, and DISI32-EW Boards, be sure to refer to the Installation Guide (Dialogic® Quick Install Card) that is provided with each board for important information about power budgeting and guidelines for selecting the slot where a board can be installed.

1.24 Support for PCI Express Boards - Dialogic® DM/V-B Boards

Summary

With the Service Update, Dialogic® System Release 6.1 for Linux now supports the following Dialogic® DM/V-B Boards in the PCI Express form factor:

Dialogic® DMV300BTEPEQ Media Board

The DMV300BTEPEQ Board is a single span board with software selectable T1/E1.

- One digital network interface with 30+ channels of media processing.
- Support for universal media load with simultaneous voice, fax, and conferencing.
- Provides A-law/mu-law conversion and the ability to mix selected protocols on the board.
- PCI Express form factor.

Dialogic® DMV600BTEPEQ Media Board

The DMV600BTEPEQ Board is a dual span board with software selectable T1/E1 (per network interface).

- Two digital network interfaces with 60+ channels of media processing.
- Support for universal media load with simultaneous voice, fax, and conferencing.
- Provides A-law/mu-law conversion and the ability to mix selected protocols on the board.
- PCI Express form factor.

Dialogic® DMV1200BTEPEQ Media Board

The DMV1200BTEPEQ Board is a quad span board with software selectable T1/E1 (per network interface).

- Four digital network interfaces with 120+ channels of media processing.
- Support for universal media load with simultaneous voice, fax, and conferencing.
- Provides A-Law/Mu-Law conversion and the ability to mix selected protocols on the board.
- PCI Express form factor.

When configuring the system for the PCI Express form factor boards, use the same menu selections and configuration settings that are documented for the PCI version of the boards. Any differences are discussed below.

Note: When installing the Dialogic® DMV300BTEPEQ, DMV600BTEPEQ, and DMV1200BTEPEQ Boards, be sure to refer to the Installation Guide (Dialogic® Quick Install Card) that is provided with each board for important information about power budgeting and guidelines for selecting the slot where a board can be installed.

Features

The Dialogic® DMV300BTEPEQ, DMV600BTEPEQ, and DMV1200BTEPEQ Boards support the same features as the existing Dialogic® DM/V-B PCI Boards plus new media loads, lower latencies/increased performance, and first time support for a single span Dialogic® DM3 Board.

Some of the key features for these boards are listed below. Refer to the product data sheet, which is accessible at <http://www.dialogic.com/products/list.asp>, for additional information about applications, configurations, features, and technical specifications.

- Software selectable T1/E1. Ability to mix T1 and E1 on each network interface.
- Ability to combine protocols on the same board. Protocols within a group can be mixed among network interfaces on the same board; however, protocols from different groups cannot be mixed on the same board.
 - Group 1: Mix any combination of 4ESS (T1), 5ESS (T1), NTT (T1), NI2 (T1), DMS (T1), QSIGT1 (T1), QSIGE1 (E1), NET5 (E1), T1CC (T1 clear channel), CAS (T1), E1CC (E1 clear channel), R2MF (E1) protocols on the same board.
 - Group 2: Mix any combination of DPNSS (E1) or DASS2 (E1) protocols on the same board.
- Ability to send alarm state to the network at all times from power-up to application start-up (i.e., trunk preconditioning).
- Universal load available (simultaneous voice + speech + fax + conferencing) on all Dialogic® DM/V-B Boards. All supported media loads are listed below.
- A-Law/Mu-Law conversion.

Note: Fixed routing configuration is not supported on Dialogic® DM/V-B Boards. Refer to the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* for more information about configuration, media loads, and mixing of protocols for the above features and board support.

Media Loads

The media loads supported by the Dialogic® DMV300BTEPEQ, DMV600BTEPEQ, and DMV1200BTEPEQ Boards are listed below.

ML	Media Loads/Features Supported											
	Voice Only								Fax	Conferencing Only		
	Basic Voice	Transaction Record	Enhanced Coders	TrueSpeech	Enhanced Echo Cancellation†	CSP	CSP Streaming to CT Bus	FSK	Fax	Conferencing Parties	Conferencing - Tone Clamping	Conferencing - Echo Cancellation
Dialogic® DMV300BTEPEQ (Single Span Board)												
UL1	30	30	30	30	30	30	30	30	12	30	30	-
ML5	30	30	-	-	-	-	-	30	30	-	-	-
ML10	30	30	30	30	30	30	30	30	-	32	32	32
Dialogic® DMV600BTEPEQ (Dual Span Board)												
UL1	60	60	60	60	60	60	60	60	16	60	60	60
UL2	90	90	90	90	90	90	90	90	6	48	48	48
Dialogic® DMV1200BTEPEQ (Quad Span Board)												
UL1	120	120	120	120	120	120	120	120	12	30	30	-
UL2	120	120	-	-	-	-	-	120	12	120	120	-
UL3	120	120	120	-	-	120*	-	120	8	36	36	36
ML2	150	150	150	150	150	150	-	150	-	-	-	-
ML5	120	120	-	-	-	-	-	120	30	-	-	-
ML5B	120	120	120	120	120	120	120	120	12	-	-	-
ML9B**	-	-	-	-	-	-	-	-	-	160	160	160
ML9C**	-	-	-	-	-	-	-	-	-	576	-	-
ML9D**	-	-	-	-	-	-	-	-	-	270	270	-
ML10	120	120	120	-	-	120*	120	120	-	54	54	54
ML10B	120	120	-	-	-	-	-	120	-	120	120	120
Notes: For more information about media loads, refer to the <i>Dialogic® DM3 Architecture Products on Linux Configuration Guide</i> . Features within a resource group (headings marked as Voice Only, Fax, or Conferencing Only) are inclusive. Features across resource groups are additive. For example, on the Dialogic® DMV600BTEPEQ Board using UL1, there are 60 total voice resources, 16 fax resources, and 60 conferencing resources. This means that any combination of the listed voice resources (Voice Only subheadings marked as Basic Voice, Transaction Record, Enhanced Coders, TrueSpeech, Enhanced Echo Cancellation, CSP, CSP Streaming to CT Bus, and FSK) can be used up to a total of 60. For example, 30 Basic Voice plus 10 Enhanced Coders plus 10 TrueSpeech plus 10 CSP Streaming to CT Bus. In addition to these various voice resources, the UL1 media load can use 16 fax resources and 60 conferencing resources (with Tone Clamping and Echo Cancellation) simultaneously. † Default configuration is EEC (enhanced EC, 32 ms) for CSP supported ML, unless otherwise indicated or set in the component named [0x2c] in the respective CONFIG file. You can only change it to a lower EC tail length, by changing the CSP parameter 0x2c03 accordingly in the respective CONFIG file. Conferencing EC, however, will always be 16 ms, regardless of the EC parameter setting. *16 ms only CSP. **There is no network interface support with these media loads. They behave as if the board is a resource-only board. Network interfaces are disabled.												

1.25 New Parameter for Adjusting Silence Threshold on Dialogic® DM3 Boards

With the Service Update, the user has the ability to adjust the silence threshold parameter on Dialogic® DM3 Boards to a value above or below the default value of -43 dBm0 while using play and record functions like **dx_play()**, **dx_record()**, and **ec_reciottdata()**. For instance, its adjustment affects the threshold for silence termination conditions in the Dialogic® R4 API TPT structure. It also affects silence detection via R4 unsolicited SRL events.

The silence threshold is the level that defines whether the incoming data to a voice channel is recognized as silence or non-silence. The threshold is defined by the minimum energy level of a signal below which it is considered as silence. With this new feature, the user can statically adjust the silence threshold default value of -43 dBm0 via the firmware configuration file across all voice channels on a Dialogic® DM3 Board.

Configuration Example

To change the default value of the silence threshold, you must add a new parameter in the CONFIG file that was selected for your board. The parameter is **0x70B**, and must be added in the [sigDet] section of the CONFIG file. A value equal to the desired silence threshold, measured in dBm0, must be entered. For example:

```
[sigDet]

SetParm=0x70B, 0xffd3 ! SD_ParmMinEnergy in dBm0 (e.g. 0xffd3=-45, 0xffda=-38,
Default: 0xffd5=-43)
```

After the CONFIG file is saved, the changes take effect after downloading.

For further information about modifying CONFIG files, see the *Dialogic® DM3 Architecture Products on Linux Configuration Guide*.

1.26 New ANI/DNIS-Enabled Parsing Tool (ADEPT) for Dialogic® PBX Integration Boards

With the Service Update, you can now use a tool called the ANI/DNIS-Enabled Parsing Tool (ADEPT), which is a set of software modules that function as a portable, generic, Calling Party Identification (CPID), Automatic Number Identification (ANI), and Dialed Number Identification Service (DNIS) display parser. By using ADEPT, a user can update display parsing rules that are specific to a site or an application.

Previous display parsers for the Dialogic® PBX Integration Boards could not be modified without rebuilding the firmware. With the ADEPT implementation, the CPID parsing is no longer controlled in the firmware. Instead, it is controlled in the d42 library with appropriate rules changes. The advantage of ADEPT is that a customer can change the display parser just by changing the rules file. The rules file is implemented as a simple text file.

ADEPT provides significant productivity advantages by eliminating the need to rebuild firmware. No recompilation is needed in any subsystem.

Both the Dialogic® D/82JCT-U and Dialogic® D/42JCT-U PBX Integration Boards support the ADEPT functionality, as well as the PCI Express versions of these boards (Dialogic® D/82JCT-EW and Dialogic® D/42JCT-EW Boards).

Detailed information about this feature is provided in a new document, *Dialogic® ADEPT for PBX Integration Boards User Guide*, which has been added to the online bookshelf for Dialogic® System Release 6.1 for Linux.

Note: Additional information for Linux and Windows® has been added to [Section 3.4.22, “Dialogic® ADEPT for PBX Integration Boards User’s Guide”](#), on page 192 in the [Documentation Updates](#) section of this Release Update.

1.27 Dynamically Changing the Transmit Time Slot on IP Media Devices

With the Service Update, the ability to dynamically change the transmit time slot on IP Media (ipmBxCx) devices is provided. The feature is supported on Dialogic® IPT Boards only.

1.27.1 Feature Description

The H.110 TDM transmit time slots for IP Media devices are allocated by the time slot allocation during the board initialization time. In previous releases, this allocation could not be changed at runtime. Starting with this Service Update, the TDM transmit time slots on the IP Media devices associated with Dialogic® IPT Boards can be dynamically changed at runtime using the **ipm_SetParm()** IP Media Library (IPML) function and a new PARMCH_TX_TIMESLOT parameter that has been created for the IPM_PARM_INFO data structure eIP_PARM define to support this feature.

The companion **ipm_GetParm()** function can be used to retrieve the transmit time slot to which the IP Media device is currently allocated. The newly assigned transmit time slot allocation remains in effect until the device is closed or another request to change the transmit time slot allocation is issued.

Note: Upon a device close (that is, **ipm_close()**), the Tx time slot for the specified device will revert back to the static allocated value assigned during board initialization.

Caution: *Assigning transmit time slots incorrectly can cause physical damage on the Dialogic® IPT Board or any other H.110 board when two or more devices are transmitting on the same TDM time slot. It is the responsibility of the application to ensure that the transmit time slot of each IP Media device is allocated to a time slot that has no other device transmit time slots allocated to it.*

1.27.2 New PARMCH_TX_TIMESLOT Parameter

A new PARMCH_TX_TIMESLOT parameter has been created for the IPM_PARM_INFO data structure eIP_PARM define to support this feature:

PARMCH_TX_TIMESLOT

Identifies a transmit time slot value for an IP Media device that can be dynamically set or retrieved at runtime.

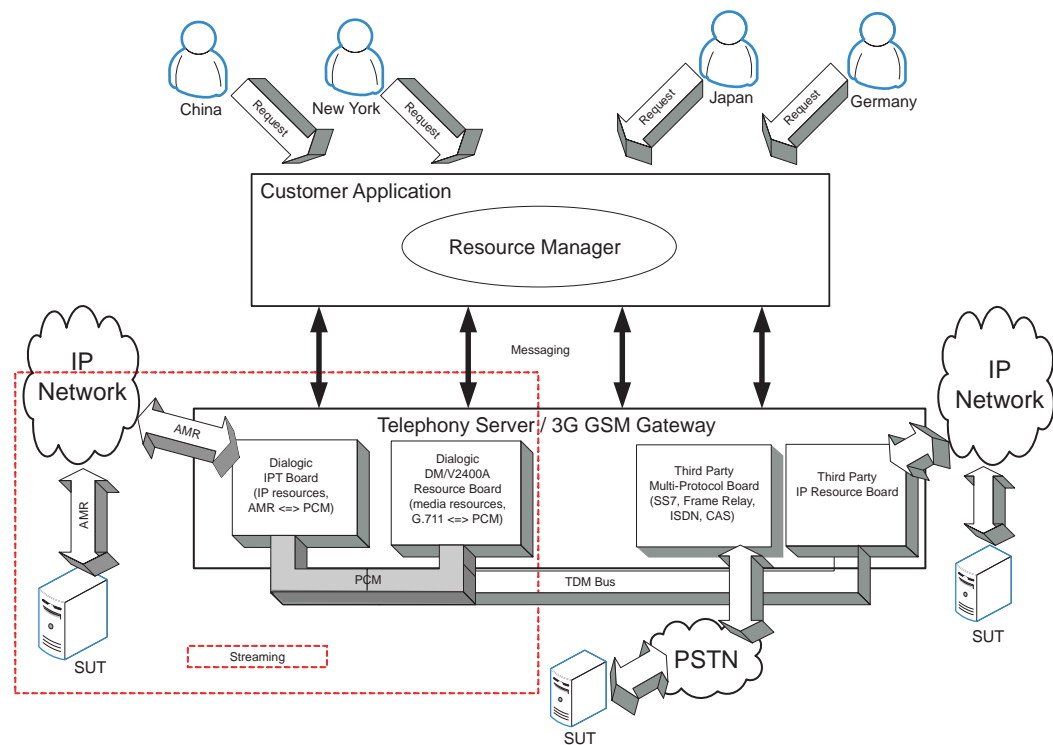
Type: unsigned short. Valid values: 0 to 4095.

1.27.3 Use Cases

A number of use case scenarios that describe applications for this feature are given in the following subsections. The system configuration for use cases 1 and 2 is described first.

System Configuration for Use Cases 1 and 2

The figure below shows the system configuration for use cases 1 and 2.



In this configuration, the customer application is designed to expect time slots in the range 1 to 240 to be assigned to the Dialogic® DM/V2400A Media Board media resources for transmitting data and time slots in the range 2001 to 2240 for receiving data (see the following table).

Boards	Static Tx TDM Time Slots	Tx TDM Time Slots	Rx TDM Time Slots	TDM Time Slot Pool
DM/V2400A	1 to 240	1 to 240	2001 to 2240	---
Third Party Boards	---	---	---	2241 to 4096

The application has an administration component that runs and creates a lookup table that is used by the Resource Manager (RM) for establishing full duplex connections between resources.

With the addition of a Dialogic® IPT Board into the system configuration, the lookup table is modified as follows:

Boards	Static Tx TDM Time Slots	Tx TDM Time Slots	Rx TDM Time Slots	TDM Time Slot Pool
DM/V2400A	1 to 240	1 to 240	2001 to 2240	---
IPT2400	241 to 480	241 to 480		
Third Party Boards	---	---	---	2241 to 4096

The time slot ranges in bold are reserved by the administration component of the application. If more Dialogic® IPT Boards are included, additional transmit time slots are assigned to the IP Media resources. Note that for the Dialogic® IPT2400 IP Board, the transmit (Tx) time slot range is 241 to 480 as statically assigned at board initialization, but these time slots can be changed at runtime using this feature.

Note: TDM Rx time slots for the Dialogic® boards are always allocated dynamically by the application.

Use Case 1

This use case describes the validation of DTMF quality sent to and received from the 3G GSM Gateway in the figure above. The application allocates a voice resource (dxxB1C1) and IP Media resource (ipmB1C100) for establishing a media session between the 3G GSM Gateway and a test tool. Using this feature, the RM changes the transmit time slot of the ipmB1C100 device (which is statically allocated to time slot #341) to time slot #2001 to match the receive time slot of the voice resource (dxxB1C1), thereby establishing a full-duplex connection between the devices as shown in the following table.

Device	Static Tx TDM Time Slot	Tx TDM Time Slot	Rx TDM Time Slot	Purpose
Voice device (dxxB1C1)	1	1	2001	Generate test DTMF Detect tones returned from SUT
IP Media device (ipmB1C100)	341	2001	1	Send/receive and encode/decode DTMF from AMR to PCM and PCM to AMR

Once the test is completed, the devices are closed and any transmit time slot allocations are freed. Devices that are subsequently opened default to the static transmit time slot assignment, unless explicitly changed using the **ipm_SetParm()** function.

Note: It is the responsibility of the application to keep track of the resource and time slot status.

This use case activities are as follows:

1. The user makes a request to set up a connection with a 3G GSM SUT to validate DTMF and audio quality.
2. The test application establishes the connection with the remote SUT using IPT (network) and IPM (media) resources.
3. The user selects the DTMF tones to send and the test application generates and sends DTMF tones to the SUT. The tones are generated using a Dialogic® DM/V2400A Media Board voice resource and G.711 encoding.
4. The DTMF is received by the IPM resource listening to transmit time slot #1 of the DM/V2400A Board voice resource.
5. The IPM resource encodes the PCM data stream to AMR and sends the RTP packet to the SUT.
6. The SUT receives the RTP stream and interprets the payload as a DTMF tone.
7. The SUT transmits the received DTMF back to the IPM resource in AMR format.
8. The IPM resource transcodes the AMR RTP stream to PCM, which is then transmitted on time slot #2001 to the DM/V2400A Board voice resource for tone detection.
9. The DM/V2400A Board voice resource signals the application that a tone has been received.

Using this scenario, the user can verify if the 3G GSM Gateway is receiving, detecting, and generating DTMF correctly.

Use Case 2

The same scenario as use case 1 above can be performed between the 3G GSM Gateway IPM and voice resources to verify voice quality. The customer can use a recorded voice file, which is sent to the SUT. Then, some form of perceptual measurement (PESQ or PSQM) can be used to evaluate the voice quality of the recorded voice received versus that which was sent.

Use Case 3

Another use case for this feature is the verification of the behavior of a specialized switch when receiving specific tones. These tests are typically run remotely to ensure the switch

is recognizing the tones and responding appropriately. The tones typically tested are call progress, SIT, and fax tones.

1.27.4 Using This Feature

The following is the sequence of function calls involved when using this feature:

1. **ipm_Open()** - Opens an IP Media device and returns a valid handle. The transmit time slot is allocated to the static time slot number.
2. **ipm_SetParm()** - Sets the transmit time slot to a specific number. This is achieved by setting up an `IPM_PARM_INFO` structure where the `eParm` field is `PARMCH_TX_TIMESLOT` and the `*pvParmValue` field is a pointer to the parameter value, that is, the time slot number to be allocated. See the **ipm_SetParm()** function reference page in the *Dialogic® IP Media Library API Library Reference* for more detail.
3. **ipm_GetParm()** (optional) - Validates that the time slot has been changed to the desired time slot. This can be used at any time to get the active transmit time slot of an IP Media device. This functionality is equivalent to that provided by the **ipm_GetXmitSlot()** function.
4. **ipm_Close()** - Closes an IPM device, deallocating all time slots. This should be the final step before the resource is returned to the free resource pool.

Caution: *Assigning transmit time slots incorrectly can cause physical damage on the Dialogic® IPT Board or any other H.110 board when two or more devices are transmitting on the same TDM time slot. It is the responsibility of the application to ensure that the transmit time slot of each IP Media device is allocated to a time slot that has no other device transmit time slots allocated to it.*

Notes: 1. This feature is applicable to Dialogic® IPT Boards only.

2. The default static time slot assignment applies if not explicitly changed using this feature.
3. It is assumed that the system is operating in Passive mode, where the customer application is responsible for the establishment and management of H.110 clocking (primary clock, signal reference, and any associated fallback strategies, respectively).
4. When a single board stop operation is performed (that is, a Dialogic® IPT Board is stopped without stopping the system), all dynamically assigned transmit time slots associated with the IP Media devices on the IPT Board are deallocated.
 - If the IPT Board is restarted without any other boards being stopped prior to the restart, the same set of static time slots are allocated to the IPT Board.
 - If multiple boards are stopped, then the static time slot allocations may change depending on the order in which the boards are restarted.
5. It is the responsibility of the application to ensure that a time slot value specified is in the range 0 to 4095. The **ipm_SetParm()** and **ipm_GetParm()** functions do not perform any validation on the value specified.

6. The **ipm_GetXmitSlot()** function, which returns the currently allocated transmit time slot for an IP Media device, has not changed as a result of this feature.
7. Closing an IP Media device, then reopening it, resets the transmit time slot back to the static time slot allocated during board initialization (firmware download).

1.27.5 Documentation

The online bookshelf provided with Dialogic® System Release 6.1 for Linux contains information about all system 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*

Note: The online bookshelf has not been updated for this feature, so the manuals above do not contain information relating to this feature.

1.28 Runtime Control of Double Answer for R2MF

With this Service Update, a connection method called double answer is now supported for rejecting collect calls on a call-by-call basis. For further information about this feature, see the *Dialogic® Global Call E1/T1 CAS/R2 Technology Guide*.

1.29 Support for Dialogic® D/4PCI Voice Board

With this Service Update, the Dialogic® D/4PCI Voice Board that was supported in older system releases is now supported in Dialogic® System Release 6.1 for Linux. In support of this board, you may now select D/4PCI from the Model Names list when using the Dialogic® Configuration Manager for Linux.

When configuring Dialogic® Springware Boards, boards that share the same Device ID with other boards are displayed in the Board Summary screen as “Name Unresolved.” When you select the thumb wheel number of the board to configure it, a menu for that board is displayed. When you choose the Model Name option, a list of possible model names for the board are displayed. D/4PCI now appears in the list with these other model names:

- D/41JCT
- VFX/41JCT
- D/4PCI-U
- D/4PCI

Detailed procedures for using the Dialogic® Configuration Manager for Linux to configure boards are given in the *Dialogic® Springware Architecture Products on Linux Configuration Guide*.

1.30 Support for PCI Express Boards - Dialogic® Springware Boards

With the Service Update, Dialogic® System Release 6.1 for Linux now supports the following PCI Express boards:

- Dialogic® D/42JCT-EW and Dialogic® D/82JCT-EW PBX Integration Boards
- Dialogic® D/240JCT-T1-EW and Dialogic® D/300JCT-E1-EW Media Boards
- Dialogic® D/480JCT-EW and Dialogic® D/600JCT-EW Media Boards
- Dialogic® D/4PCIE-4S-W and Dialogic® D/4PCIE-4F-W Media Boards
- Dialogic® D/41JCT-LS-EW and Dialogic® VFX/41JCT-LS-EW Media Boards
- Dialogic® D/120JCT-LS-EW Media Board

When configuring the system for the PCI Express form factor boards, use the same menu selections and configuration settings that are documented for the PCI version of the boards. Any differences are discussed below.

Dialogic® D/42JCT-EW and Dialogic® D/82JCT-EW PBX Integration Boards

The Dialogic® D/42JCT-EW and Dialogic® D/82JCT-EW PBX Integration Boards offer advanced digital connectivity to many of today's most popular private branch exchanges (PBXs) for unified and Internet-ready call, voice, and fax processing in small- to medium-sized enterprises. The D/42JCT-EW Board is a 4-port voice processing board in a full-length PCI Express form factor. The D/82JCT-EW Board is an 8-port voice processing board in a full-length PCI Express form factor.

Note: When installing the D/42JCT-EW and D/82JCT-EW Boards, be sure to refer to the Installation Guide (Dialogic® Quick Install Card) that is provided with each board for important information about power budgeting and guidelines for selecting the slot where a board can be installed.

Dialogic® D/240JCT-T1-EW and Dialogic® D/300JCT-E1-EW Media Boards

The Dialogic® D/240JCT-T1-EW Media Board is a 24-channel voice and T1 network interface board in a full-length PCI Express form factor.

The Dialogic® D/300JCT-E1-EW Media Board is a 30-channel voice and E1 network interface board in a full-length PCI Express form factor. The board is available in a 75-Ohm version and a 120-Ohm version.

Note: When installing the D/240JCT-T1-EW and D/300JCT-E1-EW Boards, be sure to refer to the Installation Guide (Dialogic® Quick Install Card) that is provided with each board for important information about power budgeting and guidelines for selecting the slot where a board can be installed.

Dialogic® D/480JCT-EW and Dialogic® D/600JCT-EW Media Boards

The Dialogic® D/480JCT-EW and Dialogic® D/600JCT-EW PCI Express form factor boards include the following models:

Dialogic® D/480JCT-1T1-EW Media Board

Provides up to 24 channels of combined media resources and a single T1 network interface in a single, full-length PCI Express slot.

Dialogic® D/480JCT-2T1-EW Media Board

Provides up to 48 channels of combined media resources and two T1 network interfaces in a single, full-length PCI Express slot.

Dialogic® D/600JCT-1E1-75-EW Media Board

Provides up to 30 channels of combined media resources and a single 75-ohm E1 network interface in a single, full-length PCI Express slot.

Dialogic® D/600JCT-1E1-120-EW Media Board

Provides up to 30 channels of combined media resources and a single 120-ohm E1 network interface in a single, full-length PCI Express slot.

Dialogic® D/600JCT-2E1-75-EW Media Board

Provides up to 60 channels of combined media resources and two, 75-ohm E1 network interfaces in a single, full-length PCI Express slot.

Dialogic® D/600JCT-2E1-120-EW Media Board

Provides up to 60 channels of combined media resources and two, 120-ohm E1 network interfaces in a single, full-length PCI Express slot.

- Notes:**
1. When installing the D/480JCT-EW and D/600JCT-EW PCI Express Boards, be sure to refer to the Installation Guide (Dialogic® Quick Install Card) that is provided with each board for important information about power budgeting and guidelines for selecting the slot where a board can be installed.
 2. The D/480JCT-EW and D/600JCT-EW PCI Express Boards can be used with any Dialogic® System Release 6.1 for Linux Service Update release; it is not necessary to upgrade to a particular Service Update.

Dialogic® D/4PCIE-4S-W and Dialogic® D/4PCIE-4F-W Media Boards

The Dialogic® D/4PCIE-4S-W and Dialogic® D/4PCIE-4F-W Media Boards are combined media analog boards with four ports of voice, fax, and speech in a half-length PCI Express form factor. The D/4PCIE-4S-W Board has four ports of voice and speech, and the D/4PCIE-4F-W Board has four ports of voice and fax.

Dialogic® D/41JCT-LS-EW and Dialogic® VFX/41JCT-LS-EW Media Boards

The Dialogic® D/41JCT-LS-EW and Dialogic® VFX/41JCT-LS-EW Media Boards are combined media analog boards with H.100 connectivity and four ports of voice, fax, and speech in a full-length PCI Express form factor. The D/41JCT-LS-EW Board supports basic fax, and the VFX/41JCT-LS-EW Board supports enhanced fax.

Dialogic® D/120JCT-LS-EW Media Board

The Dialogic® D/120JCT-LS-EW Media Board is a 12-port analog telecom board in a full-length PCI Express form factor. The D/120JCT-LS-EW Board supports voice, fax, and software-based speech recognition processing in a single PCI Express slot, and provides 12 analog telephone interface circuits for direct connection to analog loop start lines.

- Notes:**
1. When installing the D/120JCT-LS-EW Board, be sure to refer to the Installation Guide (Dialogic® Quick Install Card) that is provided with each board for important information about power budgeting and guidelines for selecting the slot where a board can be installed.
 2. The D/120JCT-LS-EW Board can be used with any Dialogic® System Release 6.1 for Linux Service Update release; it is not necessary to upgrade to a particular Service Update.

1.31 Support for 12 GB RAM

With this Service Update, Dialogic® System Release 6.1 for Linux now supports 12 GB RAM.

1.32 Support for Reporting Billing Type

With this Service Update, for Dialogic® DM3 Boards, there is now a way for the application to know which billing type (for a call on PDK R2 protocol) was received when the lines are available for call establishment. B tones are sent to indicate whether the line is available, and also to indicate the type of billing for the call (for example, CHARGE, NO CHARGE, or CHARGE WITH CLEARING FROM INBOUND).

This feature is already supported on Dialogic® Springware Boards; however, CHARGE WITH CLEARING FROM INBOUND is a new billing type that is also supported on Springware Boards now.

For further information about this feature, see the description of the **gc_GetCallInfo()** function CALLINFOTYPE info_id parameter in the *Dialogic® Global Call E1/T1 CAS/R2 Technology Guide*.

1.33 Dynamically Adding and Deleting SS7 Circuit Groups

With the Service Update, the user can program a Dialogic® Global Call SS7 application to dynamically add new and delete existing circuit groups at runtime. Previously, the Global Call SS7 library only supported static configuration, which meant that all parameters set in the configuration files (*config.txt* and *gcss7.cfg*) could not be modified using Global Call APIs once the system started. By being able to add and delete circuit groups dynamically, the user can start with an initial number of trunks in the system, and then enable more trunks gradually without restarting the system and application. The user also has the option to delete circuit groups.

For further information about this feature, see the *Dialogic® Global Call SS7 Technology Guide*.

1.34 Dialogic® Global Call API Support for Time Slots on Dialogic® SS7 Boards Running in DTI Mode

With the Service Update, the Dialogic® Global Call API works with Dialogic® SS7 Boards that include trunks not configured for SS7 signalling (DTI mode); i.e., all the time slots on these trunks operate in clear channel mode.

For further information about this feature, see the *Dialogic® Global Call SS7 Technology Guide*.

1.35 Additional Voice Channels on Clear Channel Media Loads

With the Service Update, customers using clear channel signaling now have basic voice support on the 31st channel. The new clear channel media loads maximize density of voice resources by fully utilizing the 31st channel on each line for Dialogic® DMV1200BTEP and Dialogic® DMV1200BTEC Media Boards. On the resource boards, Dialogic® DM/V2400A-cPCI and Dialogic® DMV4800BC, new media loads with additional voice channels can be used as resources for PSTN boards in the systems that are configured for clear channel.

1.35.1 Feature Description

Media loads are pre-defined sets of features supported by Dialogic® DM3 Boards. A media load consists of a configuration file set (PCD, FCD, and CONFIG files) and associated firmware loads that are downloaded to each board. In most cases, the PCD/FCD/CONFIG file names indicate the associated media load and protocol. See the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* for more information about media loads and configuration file sets.

The new media loads are listed in the following table:

Dialogic® Board	Media Load	PCD File Name	Basic Voice
DMV1200BTEP	1	gml1_qsb_4_e1cc*	124
DMV1200BTEC	1	gml1_cpqi_sb_4_e1cc*	124
DM/V2400A-cPCI	1	ml1_cp_cires_cc.pcd	248
DMV4800BC	1	ml1_cp_ciresb_cc.pcd	496
*These media loads are created using the trunk configuration utility. This new configuration can only run with ML1; none of the other available loads for the boards are allowed. Also, the ML1 media loads support basic voice only; FSK is not supported.			

- Notes:**
1. On the Dialogic® DMV1200BTEP and Dialogic® DMV1200BTEC Boards, all spans must run clear channel; i.e., only clear channel is available during trunk configuration. Mixing of PDK/ISDN/clear channel is not permitted.
 2. The resource boards media loads are meant to run with PSTN boards running clear channel.
 3. On Dialogic® DMV2400A-cPCI Boards, transaction record is not supported at full density; if transaction record is required, only 128 voice channels on the board can run (transaction record), while all other channels must remain idle due to timeslot limitations.
 4. On Dialogic® DMV4800BC Boards, transaction record is not supported at all.
 5. On the Dialogic® DMV1200BTEP and Dialogic® DMV1200BTEC Boards, transaction record can be run at full density.

1.35.2 Supported Boards

The following boards support this feature:

- Dialogic® DMV1200BTEP Media Boards
- Dialogic® DMV1200BTEC Media Boards
- Dialogic® DM/V2400A Media Boards
- Dialogic® DMV4800BC Media Boards

1.35.3 Documentation

The online bookshelf provided with Dialogic® System Release 6.1 for Linux contains information about all system release features including features for application development, configuration, administration, and diagnostics.

For detailed information about configuring the supported boards, see the *Dialogic® DM3 Architecture Products on Linux Configuration Guide*.

1.36 Media Channel Reset Capability (Stuck Channel Recovery)

With the Service Update, whenever a media channel gets into a “stuck” state, there is a way to recover that channel without having to redownload the board or restart the application.

Note: A stuck channel is defined as a failure where the host application is unable to recover the channel and no further media operations are possible on that channel until the board is redownloaded or (in some cases) the application is restarted.

1.36.1 Feature Description

In high-density applications, media channels sometimes become stuck and no further processing can take place until the board is redownloaded or (in some cases) until the application is restarted.

This feature provides new API functions in the Dialogic® Voice library and in the Dialogic® Continuous Speech Processing (CSP) library that enable the application to recover from the stuck channel and return it to an idle and usable state.

Note: Not all stuck channels are recoverable. Also, not all errors are stuck channel errors. See [Section 1.36.3, “Restrictions and Limitations”](#), on page 89 for more information.

Supported Boards

All Dialogic® Media Span Boards support this media channel reset feature, namely Dialogic® DM/V, DM/V-A, DM/V-B, and DM/IP Boards.

New APIs

The two new API functions are:

- **dx_reset()** - Call this API to recover the media channel when the channel is stuck and in a recoverable state. If the channel is recovered, a TDX_RESET event is generated to the application, which enables the application to reuse the channel for more media functions. If the channel is not in a recoverable state, a TDX_RESETERR event is sent back to the application indicating that the specific channel is not recoverable.
- **ec_reset()** - Call this API to recover the CSP channel when the channel is stuck and in a recoverable state. If the channel is recovered, TDX_RESET and TEC_RESET events are generated to the application, which enables the application to reuse the channel for more media functions. If the channel is not in a recoverable state, TDX_RESETERR and TEC_RESETERR events are sent back to the application indicating that the specific channel is not recoverable. Note that the **ec_reset()** function resets both the voice and the CSP channels.

Function reference information is provided next.

dx_reset()

Name: dx_reset (chdev, mode)

Inputs: int chdev • valid channel device handle
 int mode • mode of operation

Returns: 0 if success
-1 if failure

Includes: `srllib.h`
`dxxxlib.h`

Category: I/O

Mode: asynchronous or synchronous

Dialogic® DM3

Platform:

■ Description

The **dx_reset()** function recovers a channel that is “stuck” (busy or hung) and in a recoverable state, and brings it to an idle and usable state. This function blocks all other functions from operating on the channel until the function completes.

Parameter	Description
chdev	Specifies the valid device handle obtained when the channel was opened using dx_open()
mode	Specifies the mode of operation: <ul style="list-style-type: none"> • EV_ASYNC – asynchronous mode. The calling thread returns immediately so it can process media functionality on other channels. • EV_SYNC – synchronous mode. The calling thread waits until the channel is recovered or discovers that the channel is not in a recoverable state.

In synchronous mode, 0 is returned if the function completes successfully, and -1 is returned in case of error.

In asynchronous mode, the TDX_RESET event is generated to indicate that the channel was recovered and is in an idle and usable state. The TDX_RESETERR event is generated to indicate that the channel is not recoverable. Issuing any other media calls on this channel will result in an error.

■ Cautions

- The **dx_resetch()** function is intended for use on channels that are stuck and not responding. Do **not** use it in place of **dx_stopch()**. Use **dx_resetch()** only if you do not receive an event within 30 seconds of when it's expected. Overuse of this function creates unnecessary overhead and may affect system performance.

■ Errors

If the function returns -1, use the Dialogic® Standard Runtime Library (SRL) Standard Attribute function **ATDV_LASTERR()** to obtain the error code or use **ATDV_ERRMSGP()** to obtain a descriptive error message. One of the following error codes may be returned:

EDX_BADPARAM
Invalid parameter

EDX_FWERROR
Firmware error

EDX_NOERROR
No error

■ Example

```
#include <srllib.h>
#include <dxxplib.h>

main()
{
    int chdev, srlmode;
    /* Set SRL to run in polled mode. */
    srlmode = SR_POLLMODE;

    if (sr_setparm(SRL_DEVICE, SR_MODEID, (void *)&srlmode) == -1) {
        /* process error */
    }

    /* Open the channel using dx_open( ). Get channel device descriptor in
    * chdev.
    */

    if ((chdev = dx_open("dxxxBlC1",NULL)) == -1) {
        /* process error */
    }

    /* continue processing */
    . .
    /* Force the channel to idle state. The I/O function that the channel
    * is executing will be terminated, and control passed to the handler
    * function previously enabled, using sr_enbhdr(), for the
    * termination event corresponding to that I/O function.
    * In asynchronous mode, dx_stopch() returns immediately,
    * without waiting for the channel to go idle.
    */

    if ( dx_stopch(chdev, EV_ASYNC) == -1) {
        /* process error */
    }

    /* Wait for dx_stopch() to stop the channel and return the termination event
    * for the present media function.
    */

    /* After waiting for 30 secs if the termination event is not returned, issue a
    * dx_resetch() to reset the channel.
    */

    if (dx_resetch(chdev, EV_ASYNC) <0 )
    {
        /*process error */
    }
}
```

```
    }  
  
    /* Wait for TDX_RESET or TDX_RESETEERR events */  
}
```

■ **See Also**

- **ec_resetch()** in the *Dialogic® Continuous Speech Processing API Library Reference*

ec_reset()

Name: ec_resetch (chdev, mode)

Inputs: int chdev

```
int mode
```

- valid channel device handle
- mode of operation

Returns: 0 if success

-1 if failure

Includes: srllib.h

eclib.h

Category: I/O

Mode: asynchronous or synchronous

Dialogic® DM3

Platform:

■ Description

The **ec_reset()** function recovers a channel that is “stuck” (busy or hung) and in a recoverable state, and brings it to an idle and usable state. This function blocks all other functions from operating on the channel until the function completes. This function recovers both the CSP channel and the voice channel.

Parameter	Description
chdev	Specifies the valid device handle obtained when the channel was opened using dx_open()
mode	Specifies the mode of operation: <ul style="list-style-type: none"> • EV_ASYNC – asynchronous mode. The calling thread returns immediately so it can process media functionality on other channels. • EV_SYNC – synchronous mode. The calling thread waits until the channel is recovered or discovers that the channel is not in a recoverable state.

In synchronous mode, 0 is returned if the function completes successfully, and -1 is returned in case of error.

In asynchronous mode, the TDX_RESET and the TEC_RESET events are generated to indicate that the channel was recovered and is in an idle and usable state. The TDX_RESETER and the TEC_RESETER events are generated to indicate that the channel is not recoverable. Issuing any other media calls on this channel will result in an error.

■ Cautions

- The `ec_resetch()` function is intended for use on channels that are stuck and not responding. Do **not** use it in place of `ec_stopch()`. Use `ec_resetch()` only if you do not receive an event within 30 seconds of when it's expected. Overuse of this function creates unnecessary overhead and may affect system performance.

■ Errors

If the function returns -1, use the Dialogic® Standard Runtime Library (SRL) Standard Attribute function **ATDV_LASTERR()** to obtain the error code or use **ATDV_ERRMSGP()** to obtain a descriptive error message. One of the following error codes may be returned:

EDX_BADPARAM
Invalid parameter

EDX_FWERROR
Firmware error

EDX_NOERROR
No error

■ Example

```
#include <stdio.h>
#include <srllib.h>
#include <dxlib.h>
#include <eclib.h>
#include <errno.h> /* include in Linux applications only; exclude in Windows */

main()
{
    int chdev, srlmode;
    /* Set SRL to run in polled mode. */
    srlmode = SR_POLLMODE;
    if (sr_setparm(SRL_DEVICE, SR_MODEID, (void *)&srlmode) == -1) {
        /* process error */
    }

    /* Open the channel using dx_open( ). Get channel device descriptor
    * in chdev.
    */
    if ((chdev = dx_open("dxxxB1C1",0)) == -1) {
        /* process error */
    }
    /* continue processing */
    . .
    /* Force the channel to idle state. The I/O function that the channel
    * is executing will be terminated, and control passed to the handler
    * function previously enabled, using sr_enbhdr(), for the
    * termination event corresponding to that I/O function.
    * In the asynchronous mode, ec_stopch() returns immediately,
    * without waiting for the channel to go idle.
    */
    if (ec_stopch(chdev, FULLDUPLEX, EV_ASYNC) == -1) {
        /* process error */
    }

    /* Wait for the termination events (TEC_STREAM and/or TDX_PLAY) */

    /* After waiting for 30 secs, if the channel is still in a busy state,
    * issue ec_reset() to reset both the CSP channel and the voice channel.
    * When issued in asynchronous mode, it will return both (TEC_RESET/TEC_RESETErr)
    * and (TDX_RESET/TDX_RESETErr) events.
    */

    if (ec_reset(chdev, EV_ASYNC) == -1) {
        /* process error */
    }
}
```

```
/* Wait for TEC_RESET/TEC_RESETEERR and TDX_RESET/TDX_RESETEERR */  
}
```

■ See Also

- **dx_resetch()** in the *Dialogic® Voice API Library Reference*

1.36.2 Implementation Guidelines

The following guidelines apply when implementing the media channel reset capability using the Dialogic® Voice API:

- It is recommended that you issue the function in asynchronous mode for more efficient processing. In synchronous mode, the calling thread is blocked until the function completes, which may take up to a minute in worst-case scenarios.
- The **dx_resetch()** function is intended for use on channels that are stuck and not responding. Do **not** use it in place of **dx_stopch()**. Use **dx_resetch()** only if you do not receive an event within 30 seconds of when it's expected. Overuse of this function creates unnecessary overhead and may affect system performance.
- If you call **dx_resetch()** immediately following **dx_stopch()** without waiting at least 30 seconds for **dx_stopch()** to complete, you will not receive events, such as TDX_PLAY and TDX_RECORD, even if the stop operation is successful and the channel was not stuck. Instead, you will only receive the TDX_RESET event if the channel recovery is successful or the TDX_RESETErr event if the channel is not recoverable.
- If you call **dx_resetch()** without first using **dx_stopch()** to stop the channel, the Voice library will internally call **dx_stopch()** and wait 30 seconds for it to complete. If the internal stop channel is successful, you will receive the TDX_RESET event only. If the internal stop channel is unsuccessful, the Voice library will then call **dx_resetch()**. Once a reset is attempted, you will receive the TDX_RESET event if the channel recovery is successful or the TDX_RESETErr event if the channel is not recoverable.
- Unrecoverable channels are written to a log file in the DebugAngel tool or the Runtime Trace Facility (RTF) tool. See the *Dialogic® System Software Diagnostics Guide* for more information on these tools.

The following guidelines apply when implementing the media channel reset capability using the Dialogic® Continuous Speech Processing (CSP) API:

- The guidelines described for **dx_resetch()** and **dx_stopch()** apply to the **ec_resetch()** and **ec_stopch()** functions in the CSP API.
- For CSP applications, it is recommended that you use **ec_resetch()** since this function resets both the voice and the CSP channels. The **dx_resetch()** function resets the voice channels only.

1.36.3 Restrictions and Limitations

The following restrictions and limitations apply to the media channel reset feature:

- This feature only addresses scenarios where the firmware and the host library have lost synchronization or an event has not been propagated. DSP crashes, catastrophic firmware failures (killtasks), or unsynchronized firmware state machines are **not** recoverable without redownload of the board.
- This feature only addresses channels that become stuck while performing play and record, tone generation, or FSK operations. It also addresses channels that become stuck during CSP play or record operations.

- This feature does **not** address reset of IP media channels on Dialogic® DM/IP Boards. It only addresses the reset of voice channels on DM/IP Boards.
- The reset may not succeed if CPU utilization on the host system is close to 100 percent. It is recommended that the CPU usage be at a reasonable level (less than 70 percent) before you attempt a channel reset.

1.36.4 Documentation

The online bookshelf provided with Dialogic® System Release 6.1 for Linux contains information about all system release features including features for application development, configuration, administration, and diagnostics.

For more information about the Dialogic® Voice API, see the following documents:

- *Dialogic® Voice API Programming Guide*
- *Dialogic® Voice API Library Reference*

For more information about the Dialogic® Continuous Speech Processing (CSP) API, see the following documents:

- *Dialogic® Continuous Speech Processing API Programming Guide*
- *Dialogic® Continuous Speech Processing API Library Reference*

1.37 Additional Supported Operating System Distributions (OSDs)

With the Service Update, the following Linux operating system distributions (OSDs) are now supported:

Note: Only 32-bit versions of Linux OSDs are supported.

- Red Hat Enterprise Linux 5 Update 2
- Red Hat Enterprise Linux 5

Note: The following notes apply when using Red Hat Enterprise Linux 5 (or Update 2):

- Dialogic® Boards do not support Advanced Configuration and Power Interface (ACPI) interrupt routing. ACPI interrupt routing has to be disabled for Dialogic® PCI bus devices with the following Linux kernel boot parameters: "pci=noacpi". Consult your OSD release guide for details about editing Linux kernel boot parameters.
- It is recommended to use the default kernel, not the XEN or PAE kernel. The XEN and PAE kernels are not supported by Dialogic® drivers. (XEN is an open source para-virtualization virtual machine monitor for the X86 Processor architecture. PAE is installed by default on machines with >4GB RAM; Dialogic® drivers and firmware do not support physical address extension.)
- If high record failures occur when using Dialogic® DM3 CompactPCI Boards with Red Hat Enterprise Linux 5, enable ramdisk with 512M size.

- Red Hat Enterprise Linux Version 4.0 Update 6 (Advanced Server)
- Red Hat Enterprise Linux Version 4.0 Update 5 (Advanced Server, Enterprise Server)
- Red Hat Enterprise Linux Version 4.0 Update 4 (Advanced Server, Enterprise Server, and Workstation)
- Red Hat Enterprise Linux Version 4.0 Update 3 (Advanced Server, Enterprise Server, and Workstation)
- Red Hat Enterprise Linux Version 4.0 Update 2 (Advanced Server, Enterprise Server, and Workstation)
- Red Hat Enterprise Linux Version 4.0 Update 1 (Advanced Server, Enterprise Server, and Workstation)
- Red Hat Enterprise Linux Version 3.0 Update 8 (Advanced Server, Enterprise Server, and Workstation)
- Red Hat Enterprise Linux Version 3.0 Update 7 (Advanced Server, Enterprise Server, and Workstation)
- SUSE Linux Enterprise Server 10 SP2

Note: The following notes apply when using SUSE Linux Enterprise Server 10 SP2:

- Only Dialogic® Springware Boards are supported. For a list of Springware Boards supported in Dialogic® System Release 6.1 for Linux, see the Supported Hardware chapter of the *Dialogic® System Release 6.1 for Linux Release Guide*. In the list of supported boards, Springware Boards are denoted with an asterisk (*) after the board name. For additional Dialogic® Springware Boards supported in the Service Update, see [Section 1.30, “Support for PCI Express Boards - Dialogic® Springware Boards”](#), on page 77.
- Dialogic® Boards do not support Advanced Configuration and Power Interface (ACPI) interrupt routing. ACPI interrupt routing has to be disabled for Dialogic® PCI bus devices with the following Linux kernel boot parameters: “pci=noacpi”. Consult your OSD release guide for details about editing Linux kernel boot parameters.
- SUSE Linux Enterprise Server 10 SP1
- SUSE Linux Enterprise Server 10

Note: The following note applies when using SUSE Linux Enterprise Server 10 (or SP1):

 - Dialogic® Boards do not support Advanced Configuration and Power Interface (ACPI) interrupt routing. ACPI interrupt routing has to be disabled for Dialogic® PCI bus devices with the following Linux kernel boot parameters: “pci=noacpi”. Consult your OSD release guide for details about editing Linux kernel boot parameters.
- SUSE Linux Enterprise Server 9 SP4
- SUSE Linux Enterprise Server 9 SP3
- SUSE Linux Enterprise Server 9 SP2
- SUSE Linux Enterprise Server 9

1.38 New Dialogic® Diagnostics Management Console

The Service Update introduces the Dialogic® Diagnostics Management Console (DMC) version 1.0. This GUI tool provides a means of quickly launching Dialogic® diagnostic utilities and viewing various log files created with those utilities.

The DMC:

- Provides a single portal for launching diagnostic tools:
 - AppMon
 - Castrace
 - Isdntrace
 - Dlgsnapshot
 - Dm3post
 - Debugangel
 - Getver
 - its_sysinfo
 - Pdktrace
 - Pstndiag
 - RTF Manager
 - StatusMon
- Supports local and remote execution of tools. Diagnostic tools are launched remotely via the standard remote control methods provided with the operating system, such as SSH or Remote Desktop.
- Lists the diagnostic logs available both locally and remotely for viewing.
- Launches appropriate viewers for displaying logged data.

For more information, refer to the *Dialogic® System Software Diagnostics Guide*. The DMC also has online help.

1.39 More Configurations for Optional Use of Sharing of Timeslot (SOT) Algorithm

With the Service Update, the ability to disable the Sharing of Timeslot (SOT) algorithm has been extended to support additional configurations:

- *ml10_dsa_ts16.pcd*
- *ml1b_dsa_ts16.pcd*
- *ml1b_qsa_ts16.pcd*
- *ml2_qsa_ts16.pcd*
- *ml5bc_dsa_ts16.pcd*

- `ml5bc_dsa_net5.pcd`

Note: For the table of supported boards and media loads, go to the website at:
<http://www.dialogic.com/support/helpweb/dxall/tnotes/legacy/2000/tn104.htm>

For more details on the original configurations, refer to [Section 1.52, “Optional Use of Sharing of Timeslot \(SOT\) Algorithm”](#), on page 100 of this Release Update.

Feature Limitations, Caveats, or Restrictions

The user must modify the configuration file to add this functionality.

1.40 Ability to Send and Receive DPNSS End to End Messages

With the Service Update, the user has the ability to send and receive the entire raw Digital Private Network Signalling System (DPNSS) end to end message (EEM) using API control on Dialogic® DM3 Boards. A generic mechanism enables the user to add DPNSS supplementary services (like single/dual channel transfer services, call diversion, and call waiting) without needing outside support for those services first. This feature is only supported on ISDN DPNSS loads.

For further information about this feature, see the *Dialogic® Global Call ISDN Technology Guide*.

1.41 Time Stamp for Tone-On/Off Events

With the Service Update, a new time stamp has been added to the existing DE_TONEON and DE_TONEOFF events. A new TN_TIMESTAMP structure has been added to the device header file *dxxxlib.h*. This time stamp is used to associate, or group, certain tones in order to detect a particular country tone made up of two or more defined tone templates.

1.41.1 Feature Description

To test the various tones from various countries, the Tone-On/Off Call Status Transition (CST) event data has been modified to add a time stamp structure to the end of the TN_INFO structure. The CST event data is obtained by calling **sr_getdatalen()** and **sr_getevtdatap()**. A new structure, TN_TIMESTAMP, is in the device header file, *dxxxlib.h*. If the event is for Tone-On, then the time stamp represents the tone-on time, and if the event is for Tone-Off, then it represents the tone-off time.

The Tone-On/Tone-Off messages are extended to add the “start time” and the “stop time,” respectively. These time stamps are used by the customer application to calculate the Tone-On/Tone-Off duration (cadence).

1.41.2 Supported Boards

The following board supports this feature:

- Dialogic® DM/V2400A Media Boards

1.41.3 Structure

TN_TIMESTAMP is as follows:

```
// Tone ON/OFF time stamp
typedef struct {

    unsigned long  tn_TimeStamp;    /* Time stamp for tone on/off event. The time stamp is in
                                     milliseconds from when the firmware was downloaded on the
                                     board. There is no co-relation to the system time. It wraps
                                     around every ~149 hours. */

} TN_TIMESTAMP;
```

Scenario

When a Tone-On CST event is received, the application gets the CST event data with the **sr_getdatalen()** and **sr_getevtdatap()** functions, as usual. The application then applies the TN_TIMESTAMP structure to the event data and obtains the time stamp of the tone-on event or tone-off event. The TN_TIMESTAMP structure is appended to the end of the TN_INFO structure. The CST event data comprises the DX_CST, TN_INFO, and TN_TIMESTAMP structures.

Sample

The following is an example for Tone-On. Tone-Off is done the same way.

```
DX_CST *datap;
TN_INFO *tonep;
TN_TIMESTAMP *tsp;
long timestamp; // time stamp in ms units

switch(sr_getevttype(ehandle))
{
    case TDX_CST:
        datap = (DX_CST *) sr_getevtdatap(ehandle);

        if (datap->cst_event == DE_TONEON)
        {
            tonep = (TN_INFO*)(datap+1); // tone structure starts at end of CST structure
            tsp = (TN_TIMESTAMP*)(tonep+1); // time stamp structure starts at end of
                                           TN_INFO structure.
            timestamp = tsp->tn_TimeStamp; // get the time stamp
        }

        break;
    .
    .
}
```

1.41.4 Documentation

The online bookshelf provided with Dialogic® System Release 6.1 for Linux contains information about all system release features including features for application development, configuration, administration, and diagnostics.

For more information about the Dialogic® Standard Runtime Library and Dialogic® Voice APIs, see the following documents:

- *Dialogic® Standard Runtime Library API Library Reference*
- *Dialogic® Standard Runtime Library API Programming Guide*
- *Dialogic® Voice API Programming Guide*
- *Dialogic® Voice API Library Reference*

Note: The online bookshelf has not been updated for this feature, so the manuals listed above do not contain information relating to this feature.

1.42 OA&M Error Cleanup

With the Service Update, OA&M (operations, administration, and maintenance) errors continue to be logged by the Runtime Trace Facility (RTF), as well as /usr/var/syslog. These errors have been made clearer, more concise, and more consistent.

1.43 New Runtime Trace Facility (RTF) Manager

The Service Update introduces the RTF Manager, a new GUI for the Runtime Trace Facility (RTF) diagnostic tool. RTF Manager allows users to easily configure logging and tracing levels. Previously, users had to manually edit the RTF configuration file.

For more information about the RTF Manager, refer to the *Dialogic® System Software Diagnostics Guide*.

1.44 New Application Monitor

The Service Update introduces the Application Monitor diagnostic tool. Users can set up the Application Monitor to monitor one or more applications and if a problem occurs, it will launch the `its_sysinfo` tool to collect data that can help in diagnosing the problem.

For more information about the Application Monitor, refer to the *Dialogic® System Software Diagnostics Guide*.

1.45 Improved Tracing and Error Reporting

With the Service Update, the errors and warnings that are propagated up the software stack have been made clearer, more concise, and more consistent.

1.46 New Media Load for Dialogic® DMV600BTEC Boards

The Service Update provides a new media load, ML FN, for the Dialogic® DMV600BTEC Media Board. This new media load provides a higher density of fax resources than other media loads for this board. ML FN provides 60 fax channels and 60 network interfaces (with tone resources) on a single board.

1.46.1 Feature Description

Predefined sets of features for Dialogic® DM3 Boards are provided in media loads. A media load consists of a configuration file set (PCD, FCD, and CONFIG files) and the associated firmware that is downloaded to the board. See the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* for more information about media loads.

The features and channel densities provided by media load ML FN are as follows:

Features Supported	Voice	Fax	Network Interfaces
Channel Density	0	60	60

Note: The 2048-pixel and 2432-pixel formats are supported only on the first 32 channels. The last 28 channels support 1728-pixel width only. It is up to the application to keep track of which channels support the 2432-pixel width. Attempting to transmit 2048-pixel or 2432-pixel formats on the last 28 channels will result in a TFX_FAXERROR. Attempting to receive 2048-pixel or 2432-pixel formats on the last 28 channels will result in a truncated image if the rcvflag is set to 1728-pixel or a TFX_FAXERROR if rcvflag is not set to 1728-pixel. If reception of a 2432-pixel page width image is attempted on the last 28 channels, the received image will be scaled to 1728 width provided the workaround below is used. If this workaround is not used, the reception of a 2432 width image will fail.

Workaround: As part of the required receive fax API call, “fx_rcvfax”, the application should also explicitly include the maximum receive width as follows: (h is the Fax handle)

First 32 channels of load:

```
fx_rcvfax( h , ..., EV_ASYNC | DF_PHASEB | DF_PHASED | DF_2432MAX )
```

Last 28 channels of load:

```
fx_rcvfax( h , ..., EV_ASYNC | DF_PHASEB | DF_PHASED | DF_1728MAX )
```

ML FN is supported for ISDN and PDK protocols, including mixed ISDN/CAS (i.e., ISDN on one trunk and CAS on the other trunk).

1.46.2 Configuring the Software

The new media load can be selected when using the Dialogic® Configuration Manager for Linux. The configuration procedure is described in detail in the *Dialogic® DM3 Architecture Products on Linux Configuration Guide*. This is done before the boards are started.

1.46.3 Documentation

The online bookshelf provided with Dialogic® System Release 6.1 for Linux contains information about all system release features including features for application development, configuration, administration, and diagnostics.

For detailed information about configuring Dialogic® DMV600BTEC Boards, see the *Dialogic® DM3 Architecture Products on Linux Configuration Guide*.

Note: The online bookshelf has not been updated for this feature, so the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* does not currently include information about media load ML FN.

1.47 Removal of IP Gateway R4 and IPML Gateway Demos

With the Service Update, the IP Gateway R4 and IPML Gateway demos have been removed from build and no longer supported in Dialogic® System Release 6.1 for Linux.

1.48 New Status Monitor (statusmon) Application

The Service Update introduces the Status Monitor (statusmon) application for Linux, which was previously available only on Dialogic® system release software on Windows® operating systems. Statusmon is a command line application with a terminal user interface (TUI). The application supports two execution modes:

- Monitoring call state
- Monitoring bit information

For more information about the statusmon application, refer to the *Dialogic® System Software Diagnostics Guide*.

Note: An enhanced version of statusmon was released in a later Service Update. The previous version of the tool is still supported. For information about the new version, see [Section 1.22, “Enhanced Diagnostics Tools”](#), on page 64.

1.49 New Version of Runtime Trace Facility (RTF) Tool

The Service Update introduces a new version of the Runtime Trace Facility (RTF) tool. The RTF tool provides a mechanism for tracing the execution path of runtime libraries that are supported by Dialogic® System Release 6.1 for Linux. Support for the RTF tool is as follows:

- `Rtftool` command is used to stop/start the RTF tool's tracing capabilities.
- Provides centralized logging for key OA&M components (OAMSYSLOG) and IP libraries.
- Run the RTF tool in preservation mode. Preservation mode allows you to save specified RTF trace information into a separate, preserved log file while the RTF engine continues to output active trace information into the default log file. The RTF engine will not overwrite, delete, or append to the preserved log file after it has been saved.
- Trace the Audio Conferencing API (DCB) library.

For more information about the RTF tool, refer to the *Dialogic® System Software Diagnostics Guide*.

1.50 New Version of Get Version (Getver) Tool

The Service Update introduces a new version of the Get Version (getver) tool. Getver outputs version information for files that are part of the Dialogic® software installation.

For more information about the getver tool, refer to the *Dialogic® System Software Diagnostics Guide*.

1.51 Telecom Subsystem Summary Tool (its_sysinfo)

The Service Update introduces a new version of the Telecom Subsystem Summary (its_sysinfo) tool. 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® System Release software, and operational logs. The new version of its_sysinfo supports a command line interface, provides more data-collecting features, and improves some existing data collecting features.

For detailed information about the its_sysinfo tool, see the *Dialogic® System Software Diagnostics Guide*. The following sections describe new features in the Service Update for its_sysinfo:

- [Disable Collection of Board Memory Dumps](#)
- [Linux Package Info Added to its_sysinfo](#)

1.51.1 Disable Collection of Board Memory Dumps

With the Service Update, a new option can be used to disable collection of board memory dumps for Dialogic® Springware Boards when *its_sysinfo* is run. The new command line option is **-d**:

```
its_sysinfo [filename.zip] [-d] [-?]
```

where:

filename.zip

Name of the file where the collected system information will be stored. Default is *its_sysinfo.zip*.

-d

Disables the collection of the board memory dumps. By default, board memory dumps for Springware Boards are collected under *its_sysinfo.htm*, which is one of the files that is included in the zip file.

-?

Shows the usage screen.

1.51.2 Linux Package Info Added to *its_sysinfo*

With the Service Update, the *its_sysinfo.htm* file now includes a Linux Package Info section at the beginning of the file. For example:

```
LinuxPackageInfo

Installed Dialogic RPM Information

DLGChdsi-6.2.0.0
Software for the Intel NetStructure(R) HDSI
Install Date: Tue 14 Nov 2006 07:36:50 PM EST
Build Host:  myhost
Build Date:  Thu 09 Nov 2006 02:15:31 PM EST

DLGCsdk-6.2.0.0
Intel(R) Dialogic(R) Software Development Kit
Install Date: Tue 14 Nov 2006 07:39:09 PM EST
Build Host:  myhost
Build Date:  Thu 09 Nov 2006 02:15:04 PM EST

...

Selections Made from Dialogic Install Menu

Intel Netstructure HDSI
Intel Netstructure DMIP
Global Call Protocols
Intel Dialogic Boards
...
```

1.52 Optional Use of Sharing of Timeslot (SOT) Algorithm

The Service Update now supports disabling the Sharing of Timeslots (SOT) algorithm for certain media loads. The SOT algorithm for Dialogic® DMV, DM/IP, and DM/V-A Boards with network interfaces maximizes the efficiency of the internal timeslots used for external transmit reference, allowing a full 120 channel density for such features as continuous speech processing and transaction record. The SOT algorithm is enabled by default, regardless of whether continuous speech processing or transaction record functionality is needed. Its use places certain constraints on an application for performing listen/unlisten functions in a specific sequence.

Note: The SOT algorithm does not apply to Dialogic® DM/V-B Boards, and it does not apply to resource-only DM/V-A and DMV/IP Boards.

For increased flexibility in application design, it is now possible to disable the SOT algorithm by adding a new parameter, **QKERNEL_DISABLE_TIMESLOT_SHARING**, to the board's CONFIG file, and then generating an updated FCD file.

For more detailed information about the SOT algorithm, guidelines for enabling or disabling the algorithm, and supported boards and media loads, see the technical note titled "Disabling the Sharing of Timeslot (SOT) Algorithm via DM3 config file change" on the Dialogic® Services and Support website at:

<http://www.dialogic.com/support/helpweb/dxall/tnotes/legacy/2000/tn104.htm>

1.53 Deprecation of Dlgsnmpd and Orbacus Init Scripts

In an upcoming Service Update build, two of the start/stop scripts previously included in the release will be removed. In general, these scripts are used by the system at boot time to start/stop the Dialogic® services and are not generally accessed by users.

/etc/init.d/orbacus

This script is used to start or stop the CORBA-based framework currently used by the Dialogic® System Release software. In an upcoming Service Update build, the need for this CORBA implementation will be eliminated, rendering the script unnecessary. No replacement will be offered for this interface.

/etc/init.d/dlgsnmpd

This script is used to start or stop the Dialogic® SNMP support. The functionality of this script has already been moved into the main Dialogic® System Release start/stop script, `/etc/init.d/ct_intel`; and in an upcoming Service Update build, the `dlgsnmpd` script will be removed. A table mapping old `dlgsnmpd` functionality to the updated `ct_intel` script is provided below so that users directly accessing the `dlgsnmpd` script can modify their implementations to use the `ct_intel` script instead.

ACTION	OLD	NEW
start Dialogic SNMP	<code>/etc/init.d/dlgsnmpd start</code>	<code>/etc/init.d/ct_intel snmpstart</code>
stop Dialogic SNMP	<code>/etc/init.d/dlgsnmpd stop</code>	<code>/etc/init.d/ct_intel snmpstop</code>

restart Dialogic SNMP	/etc/init.d/dlgcsnmpd restart	/etc/init.d/ct_intel snmprestart
check Dialogic SNMP status	/etc/init.d/dlgcsnmpd status	/etc/init.d/ct_intel snmpstatus

The status returned by the `ct_intel snmpstatus` command has been enhanced to show all of the three programs involved in the SNMP support. Previously, `dlgcsnmpd` only displayed the status of one of the three.

The elimination of these two scripts will result in a single point of start/stop control for all of the Dialogic® System Release software and eliminate certain issues that occurred when one of the three services was stopped without the other two being stopped.

If you select the SNMP item from the install menu, then the SNMP support will be started at system boot or when `dstart` is executed. If you do not want the SNMP support to be started:

1. Do not select “All” or “SNMP Component Manager” from the install menu.
2. Do not install the `net-snmp` RPMs on your system. Dialogic® SNMP support will not start if this dependency is not found.
3. Do not configure SNMP during the Dialogic® System Release configuration process.

1.54 Removal of Unused Header Files

With the Service Update, the following header files that were included in previous Dialogic® System Release 6.1 for Linux builds are being removed:

- `ICTDm3Board.h`
- `ICTPmacBoard.h`
- `ICTPreConfigMgr.h`
- `ICTSprwBoard.h`

These header files were inadvertently included in previous builds and are being removed since the files have no functional implementation. The removal has no functional impact on any runtime functionality that is supported for management applications built on the existing DASI implementation. If these files were included in any make rules for management applications, the associations to these header files need to be removed to avoid inadvertent compile-time errors.

1.55 Dynamic Detection of ETSI FSK Protocols

The Service Update adds a new mode for ETSI-compliant frequency shift keying (FSK) signaling in `dx_setparm()` in the Dialogic® Voice API library. This mode is used in short message service (SMS) applications, and enables Dialogic® DM3 Boards to dynamically switch between ETSI Protocol 1 and ETSI Protocol 2 as needed to complete the SMS transmission.

This new functionality is supported on Dialogic® DM/V-A and DM/V-B Media Boards.

1.55.1 Feature Description

The Dialogic® Voice API library currently supports both ETSI Protocol 1 and ETSI Protocol 2 for FSK signaling in short message service (SMS) applications. Protocol 1 is primarily for SMS applications using GSM with mobile devices. Protocol 2 is primarily for SMS applications using PSTN or ISDN in fixed residential networks.

The ETSI Protocol 1 or ETSI Protocol 2 is specified on a channel basis using the channel seizure and mark length parameters in **dx_setparm()**: DXCH_FSKCHSEIZURE and DXCH_FSKMARKLENGTH. These parameters allow the protocol to be configured in a static way between a calling party and a called party. This static definition works in an environment where the protocol to be used in an SMS transmission is known.

In some environments, however, the protocol used by a called party (ETSI Protocol 1 or ETSI Protocol 2) is unknown to the Dialogic® board. In these cases, the Dialogic® board needs to dynamically detect the protocol used by the called party in each SMS session, adjust its configuration to match the protocol in use, and successfully complete the SMS transmission.

The following new mode for DXCH_FSKSTANDARD in **dx_setparm()** is provided to enable the ETSI protocol auto-detect feature: DX_FSKSTDETSIAUTO.

The updated description for DXCH_FSKSTANDARD is as follows:

DXCH_FSKSTANDARD

Specifies the FSK protocol standard used for transmission and reception of FSK data. The protocol standard can be set to one of the following:

- DX_FSKSTDBELLCORE – Bellcore standard (default value)
- DX_FSKSTDETSI – ETSI standard
- DX_FSKSTDETSIAUTO – Auto-detect ETSI Protocol 1 or ETSI Protocol 2

If you specify DX_FSKSTDETSIAUTO, it is recommended that you call **dx_RxlottData()** to let the firmware determine the specific values for the DXCH_FSKCHSEIZURE and DXCH_FSKMARKLENGTH. The values are set based on the results of the protocol negotiation with the called party.

- Notes:**
- 1.** The ETSI auto-detect mode (DX_FSKSTDETSIAUTO) applies only in cases where the short message service center (computer where Dialogic® boards reside) is the calling party; that is, the short message service center receives the Data Link Establish message (DLL_SMS_EST). For sessions where the short message service center is the called party (that is, the short message service center sends the DLL_SMS_EST message), the application is responsible for establishing the proper FSK standard, channel seizure, and mark length values. In this case, if the wrong parameters are set, the protocol will time out and it is up to the application to retry the transmission.
 - 2.** In ETSI auto-detect mode, the channel seizure and mark length values retrieved for a called party remain in effect for subsequent sessions on the same channel, unless modified using **dx_setparm()**.

1.55.2 Using the DX_FSKSTDETSIAUTO Mode

The following steps briefly describe how to enable and use ETSI auto-detect mode in an SMS application:

1. After opening a channel device, call **dx_setparm()** and specify the ETSI auto-detect mode, DX_FSKSTDETSIAUTO, for DXCH_FSKSTANDARD.
2. Call **dx_RxlottData()** to receive FSK data on the specified channel. The firmware determines the appropriate values for channel seizure and mark length based on the results of the protocol negotiation with the called party.
3. After **dx_RxlottData()** has completed, you can call **dx_getparm()**, if desired, to retrieve the channel seizure and mark length values and determine whether the called party uses ETSI Protocol 1 or ETSI Protocol 2. Note that this step is not necessary to complete the current SMS session as the firmware will automatically detect the channel seizure and mark length values.

1.55.3 Documentation

For more information about sending and receiving FSK data, see the *Dialogic® Voice API Programming Guide*. For details about **dx_setparm()** and other Dialogic® Voice API library functions, see the *Dialogic® Voice API Library Reference*. For details about the ETSI protocols, see the ETSI ES 201 912 specification.

1.56 dx_stopch() EV_NOSTOP Mode Support for Dialogic® DM3 Boards

With the Service Update, the **dx_stopch()** function EV_NOSTOP mode can now be used with Dialogic® DM3 Boards. The **dx_stopch()** function forces termination of currently active I/O functions on a specified channel. It forces a channel in the busy state to become idle. When issued on a channel that is already idle and EV_NOSTOP is specified, the application receives a TDX_NOSTOP event to indicate that the **dx_stopch()** function completed and that no STOP was needed (that is, the channel was already idle when **dx_stopch()** was issued). Previously, no event was returned in this situation for DM3 Boards.

1.56.1 Feature Description

EV_NOSTOP can be ORed with any of the other mode flags supported by **dx_stopch()**, for example:

```
dx_stopch(chdev, EV_ASYNC|EV_NOSTOP)
```

dx_stopch() with EV_NOSTOP operates as follows:

- If issued on an idle channel, TDX_NOSTOP is returned.

- If issued on an active channel, the active channel is stopped and a media termination event (e.g., TDX_PLAY) is returned with termination reason TM_USRSTOP. (TDX_NOSTOP is not returned.)
- If issued in synchronous mode (EV_SYNC | EV_NOSTOP), no event is sent to application. **dx_stopch()** will return only when the channel comes to idle state.

Note that it is possible to get both events (TDX_PLAY and TDX_NOSTOP) in a race condition, that is, if **dx_stopch()** is issued for a channel that transitioned to idle just before the stop request, before the application received the TDX_PLAY event. In this case, the termination reason for the TDX_PLAY is **not** TM_USRSTOP.

1.56.2 Documentation

The online bookshelf provided with Dialogic® System Release 6.1 for Linux contains information about all system release features including features for application development, configuration, administration, and diagnostics.

For more information about the **dx_stopch()** function and the Dialogic® Voice API, see the following documents:

- *Dialogic® Voice API Programming Guide*
- *Dialogic® Voice API Library Reference*

1.57 Supported Kernel and GCC Versions

With the Service Update, the newly supported OSDs (Red Hat Enterprise Linux Version 4.0 and SUSE Linux Enterprise Server 9) are based on the 2.6 kernel. The Service Update will support both 2.6 and 2.4 kernel based OSDs. Redundant Host and Peripheral Hot Swap will not be supported with the 2.6 kernel in the current Service Update.

All of the supported OSDs do not use the same version of GCC, so the Dialogic® System Release 6.1 for Linux Service Update CD includes RPMs for two different versions of GCC: 3.2 and 3.4. For more information, refer to the *Dialogic® System Release 6.1 for Linux Software Installation Guide*. The *Dialogic® System Release 6.1 for Linux Release Guide* contains the list of OSDs supported for the original release as well as additional details about system requirements.

1.58 New Media Loads for Dialogic® DMV1200BTEC Boards

The Service Update provides new media loads for the Dialogic® DMV1200BTEC Media Boards.

ML QSB-U2

Provides increased density for conferencing (with echo cancellation and tone clamping), while also providing basic voice, FSK, transaction record, and fax.

ML 10B

Provides basic voice, FSK, transaction record, and conferencing (with echo cancellation and tone clamping), with channel densities designed for use with high density background music applications.

Background music is an application where two callers chat while music plays in the background. In essence, it is a 3-party conference with the third party being a voice resource playing music. Background music applications require a ratio of 2 network interfaces to 1 voice resource to 3 conference parties. The network interfaces on the Dialogic® DMV1200BTEC Board can be used, thus providing a one-board solution.

Note: ML 10B is only supported for ISDN protocols.

1.58.1 Feature Description

Predefined sets of features for Dialogic® DM3 Boards are provided in media loads. A media load consists of a configuration file set (PCD, FCD, and CONFIG files) and the associated firmware that is downloaded to the board. See the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* for more information about media loads.

Media Load QSB-U2

The features and channel densities provided by media load QSB-U2 are as follows:

Features Supported	Basic Voice (ML1), FSK, and Transaction Record	Fax	Rich Conferencing (ML9B), with Echo Cancellation and Tone Clamping
Channel Density	120	12	120

There are 120 total voice resources. Any combination of the voice features (basic voice, FSK, and transaction record) can be used up to a total of 120. There are no limitations in transaction record density supporting it at full density (120). In addition to the voice resources, 12 fax resources and 120 conferencing resources (with echo cancellation and tone clamping) can be used concurrently.

Conference size is limited to 20 parties without bridging. Conference bridging can be used to effectively expand a conference beyond the maximum size. Conference bridging consumes conferencing resources, reducing overall board conference density.

Media Load 10B

The features and channel densities provided by media load 10B are as follows:

Features Supported	Basic Voice (ML1), FSK, and Transaction Record	Rich Conferencing (ML9B), with Echo Cancellation or Tone Clamping
Channel Density	60	180

There are 60 total voice resources. Any combination of the voice features (basic voice, FSK, and transaction record) can be used up to a total of 60. There are no limitations in

transaction record density supporting it at full density (60). In addition to the voice resources, 180 conferencing resources (with echo cancellation and tone clamping) can be used concurrently.

Conference size is limited to 18 parties without bridging. Conference bridging can be used to effectively expand a conference beyond the maximum size. Conference bridging consumes conferencing resources, reducing overall board conference density. This is not an issue for background music applications, which use 3-party conferences.

Note: ML 10B can be used with any ISDN protocol supported by the Dialogic® DMV1200BTEC Board, but it cannot be used with CAS and R2MF protocols.

1.58.2 Configuring the Software

The new media loads can be selected when using the Dialogic® Configuration Manager for Linux. The configuration procedure is described in detail in the *Dialogic® DM3 Architecture Products on Linux Configuration Guide*. This is done before the boards are started.

In addition, the following runtime parameter changes are recommended when playing music into a conference (e.g., when using ML 10B):

- When you add music to the conference, set its party attributes so that it uses transmit-only mode. That is, for the conference party that transmits music, enable the MSPA_MODEXMITONLY attribute in the MS_CDT data structure chan_attr field.
- The Conferencing AGC Noise Level Lower Threshold may have to be adjusted. This parameter is set at -40 dB by default and filters out any signals below this level. If the background music levels are low, the music may not be summed into the conference or it may come in and out. In this case, the AGC Noise Level Lower Threshold should be reduced by adding “setparm=0x3b1f, <value>” to the [0x3b] section of the CONFIG file and re-downloading the board. For further information about this parameter, refer to the discussion of [CSUMS_AGC_low_threshold \(AGC Noise Level Lower Threshold\)](#) in [Section 1.66, “New Media Load for Dialogic® DMV4800BC Boards”](#), on page 111.

1.58.3 Documentation

The online bookshelf provided with Dialogic® System Release 6.1 for Linux contains information about all system release features including features for application development, configuration, administration, and diagnostics.

For detailed information about configuring Dialogic® DMV1200BTEC Boards, see the following document:

- *Dialogic® DM3 Architecture Products on Linux Configuration Guide*

For information about the Audio Conferencing (DCB) API, see the following documents:

- *Dialogic® Audio Conferencing API Programming Guide*

- *Dialogic® Audio Conferencing API Library Reference*

Note: The online bookshelf has not been updated for this feature, so the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* does not currently include information about media loads QSB-U2 and 10B.

1.59 GSM-AMR-NB Coder Support for Dialogic® IPT Boards

With the Service Update, Groupe Systeme Mobile (GSM) Adaptive Multi-Rate (AMR) Narrow Band (NB) coders are supported on Dialogic® IPT Boards. Any AMR-NB allowable rate of 4.75 to 12.2 kbps may be used. Full density of AMR coders (up to 480 channels) is supported.

Note: In order to support this feature, the IPT Board must have a revision D0 Digital Signal Processor (DSP). Older revision boards do not support this feature. To determine if your board can use the AMR-NB coders, check the Configured Assembly (CA) label, which is located on the same side of the board as the four MAC address labels. (It is in close proximity to the front panel and parallel to it.) Your board must have one of the CA numbers listed in the following table to use the AMR-NB coders.

Board Model	CA Number
Dialogic® IPT6720C	C48123-001 or C48123-003
Dialogic® IPT4800C	C56297-003
Dialogic® IPT2400C	C56004-003
Dialogic® IPT1200C	C52890-001 or C52890-003

1.59.1 Feature Description

Coder information can be specified using the IP Media Library API.

1.59.1.1 Using the Dialogic® IP Media Library API to Specify the Coder

The **ipm_StartMedia()** function in the Dialogic® IP Media Library API sets media properties and starts a media session. Coder information is provided in the IPM_MEDIA_INFO data structure. The IPM_MEDIA_INFO structure references IPM_MEDIA, which in turn references IPM_CODER_INFO for coder information.

The following new values are supported for the eCoderType field of the IPM_CODER_INFO structure:

- CODER_TYPE_AMRNB_4_75k - GSM-AMR-NB 4.75 kbps
- CODER_TYPE_AMRNB_5_15k - GSM-AMR-NB 5.15 kbps
- CODER_TYPE_AMRNB_5_9k - GSM-AMR-NB 5.9 kbps
- CODER_TYPE_AMRNB_6_7k - GSM-AMR-NB 6.7 kbps
- CODER_TYPE_AMRNB_7_4k - GSM-AMR-NB 7.4 kbps

- CODER_TYPE_AMRNB_7_95k - GSM-AMR-NB 7.95 kbps
 - CODER_TYPE_AMRNB_10_2k - GSM-AMR-NB 10.2 kbps
 - CODER_TYPE_AMRNB_12_2k - GSM-AMR-NB 12.2 kbps
- Note:** These coder types are supported only on Dialogic® IPT Boards.

When using any of these coders:

- Coder frame size is fixed at 20 ms.
- Frames per packet (fpp) is:
 - - 1 or 2 (transmit)
 - - 1, 2, or 3 (receive)
- VAD must be disabled.

1.59.2 Documentation

The online bookshelf provided with Dialogic® System Release 6.1 for Linux contains information about all system 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*

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>.

1.60 Tcl/Tk No Longer Required for QScript Utilities Support

With the Service Update, an independent Tcl/Tk package no longer needs to be installed because it has been incorporated as a component of the Dialogic® System Release 6.1 for Linux build. Also, no symbolic links are required to simulate Tcl/Tk 8.3.

1.61 System Logging Integrated with Runtime Trace Facility (RTF)

With the Service Update, the System Logging facility has been integrated with Runtime Trace Facility (RTF). The environment DLG_TRACE_LEVEL is replaced by the RTF module OAMSYSLOG. The *dlgsyslogger.log* and *oam.log* files are no longer generated

and have now been replaced by RTF log files. Refer to the *Dialogic® System Software Diagnostics Guide* for more details on RTF logging.

1.62 Support for Multiple Dialogic® SS7 Boards in the Same System

With the Service Update, Dialogic® Global Call SS7 software supports multiple Dialogic® SS7 Boards in the same system. Previous versions of Global Call SS7 supported operation with a single Dialogic® SS7 Board in the host system. Starting with the Service Update, multiple local Dialogic® SS7 Boards can be configured and used under Global Call SS7 control.

For further information about this feature, see the *Dialogic® Global Call SS7 Technology Guide*.

1.63 Support for Dialogic® DM/IP Boards and Features

The following Dialogic® DM/IP Boards are supported with the Service Update:

Dialogic® DM/IP241-1T1-PCI-100BT IP Board

24 channels of IP gateway functionality and voice processing with single T1 span and one 100 BaseT Ethernet interface; Universal PCI form factor.

Dialogic® DM/IP301-1E1-PCI-100BT IP Board

30 channels of IP gateway functionality and voice processing with single E1 span and one 100 BaseT Ethernet interface; Universal PCI form factor.

Dialogic® DM/IP481-2T1-PCI-100BTIP Board

48 channels of IP gateway functionality and voice processing with dual T1 span and one 100 BaseT Ethernet interface; Universal PCI form factor.

Dialogic® DM/IP601-2E1-PCI-100BT IP Board

60 channels of IP gateway functionality and voice processing with dual E1 span and one 100 BaseT Ethernet interface; Universal PCI form factor.

Dialogic® DM/IP481-2T1-cPCI-100BTIP Board

48 channels of IP gateway functionality and voice processing with dual T1 span and one 100BaseT on-board data network interface; single slot; cPCI form factor.

Dialogic® DM/IP601-2E1-cPCI-100BT IP Board

60 channels of IP gateway functionality and voice processing with dual E1 span and one 100BaseT on-board data network interface; single slot; cPCI form factor.

Dialogic® DM/IP601-cPCI-100BT IP Board

60 channels of IP gateway functionality and voice processing with one 100 BaseT on-board data network interface; single slot; cPCI form factor.

The Dialogic® DM/IP Boards offer zero, one, or two T1 or E1 spans plus Voice over IP (VoIP) and media processing in a single-slot solution. These boards are ideal for larger enterprise and carrier-grade IP media gateway solutions. They support the H.323 and SIP

IP telephony standards, as well as traditional APIs including R4 and Dialogic® Global Call API. For more information about IP features supported on these boards, refer to the *Dialogic® Global Call IP Technology Guide*.

1.64 Support for Redundant Host (RH)

With the Service Update, Redundant Host (RH) is now supported on the following chassis/Single Board Computers (SBC):

- Intel NetStructure ZT5085 or MPCHC5085 with MPCBL5524A1C (Dual CPU with up to 4 GB RAM)
- Intel NetStructure ZT5085 or MPCHC5085 with MPCBL5524A1D (Single CPU with up to 4 GB RAM)
- Intel NetStructure ZT5085 or MPCHC5085 with ZT5524A-1A (Dual CPU)
- Intel NetStructure ZT5085 or MPCHC5085 with ZT5524A-1B (Single CPU)

Note: This release supports CompactPCI Active/Standby Redundant Host capability. The restart mode for the components is “warm”, which means that the operating system is loaded but the software must be downloaded before the system can resume operation.

To use RH capabilities with the supported compute platforms, install the latest version of the RH software when you install the Dialogic® System Release 6.1 for Linux Service Update. For information about installing the RH software, refer to the *Pigeon Point Systems Linux Hot Swap Kit User Guide* located in the *redistributable-runtime/pps* directory of the release build. For information about using RH software, refer to the *Dialogic® System Release 6.1 for Linux Software Installation Guide*, *Dialogic® System Release 6.1 for Linux Administration Guide*, and *Dialogic® High Availability for Linux Demo Guide*.

1.65 Support for Peripheral Hot Swap (PHS) on Additional Compute Platforms

With the Service Update, Peripheral Hot Swap (PHS) is now supported on the following chassis/Single Board Computers (SBC):

- Intel NetStructure ZT5085 or MPCHC5085 with MPCBL5524A1C (Dual CPU with up to 4 GB RAM)
- Intel NetStructure ZT5085 or MPCHC5085 with MPCBL5524A1D (Single CPU with up to 4 GB RAM)

These are in addition to the following chassis/SBC that supported PHS prior to the Service Update:

- Intel NetStructure ZT5085/ZT5524A-1A (Dual CPU)
- Intel NetStructure ZT5085/ZT5524A-1B (Single CPU)
- Intel NetStructure MPCHC5091/ZT5524A-1A

- Intel NetStructure MPCHC5091/ZT5524A-1B
- Advantech MIC-3081/MIC-3369
- Advantech MIC-3038/MIC-3358
- Advantech MIC-3041/MIC-3389

To use PHS capabilities with the supported compute platforms, install the latest version of the Hot Swap Kit (HSK) software when you install the Dialogic® System Release 6.1 for Linux Service Update. For information about installing the PHS software, refer to the *Pigeon Point Systems Linux Hot Swap Kit User Guide* located in the *redistributable-runtime/pps* directory of the release build. For information about using RH software, refer to the *Dialogic® System Release 6.1 for Linux Software Installation Guide*, *Dialogic® System Release 6.1 for Linux Administration Guide*, and *Dialogic® High Availability for Linux Demo Guide*.

1.66 New Media Load for Dialogic® DMV4800BC Boards

The Service Update provides a new media load, ML10C, for the Dialogic® DMV4800BC Media Board. This new media load provides basic voice, FSK, transaction record, and basic conferencing, with channel densities designed for use with high density background music applications.

Background music is an application where two callers chat while music plays in the background. In essence, it is a 3-party conference with the third party being a voice resource playing music. Background music applications require a ratio of 2 network interfaces to 1 voice resource to 3 conference parties. (The network interfaces are not on the Dialogic® DMV4800BC Board; they have to come from another board, e.g., a Dialogic® DMT160TEC Digital Telephony Interface Board.) Echo cancellation is needed in some cases but not needed in others. ML10C is for applications that do **not** require echo cancellation.

1.66.1 Feature Description

Predefined sets of features for Dialogic® DM3 Boards are provided in media loads. A media load consists of a configuration file set (PCD, FCD, and CONFIG files) and the associated firmware that is downloaded to the board. See the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* for more information about media loads.

The features and channel densities provided by media load ML10C are as follows:

Features Supported	Basic Voice (ML1), FSK, and Transaction Record	Basic Conferencing (ML9C), No Echo Cancellation or Tone Clamping
Channel Density	120	360

There are 120 total voice resources. Any combination of the voice features (basic voice, FSK, and transaction record) can be used up to a total of 120. There are no limitations in transaction record density supporting it at full density (120). In addition to the voice

resources, 360 conferencing resources can be used concurrently (without echo cancellation or tone clamping).

Conference size is limited to 60 parties without bridging. Conference bridging can be used to effectively expand a conference beyond the maximum size. Conference bridging consumes conferencing resources, reducing overall board conference density. This is not an issue for background music applications, which use 3-party conferences.

1.66.2 Configuring the Software

The new media load can be selected when using the Dialogic® Configuration Manager for Linux. The configuration procedure is described in detail in the *Dialogic® DM3 Architecture Products on Linux Configuration Guide*. This is done before the boards are started.

In addition, the following runtime parameter changes are recommended when playing music into a conference:

- When you add music to the conference, set its party attributes so that it uses transmit-only mode. That is, for the conference party that transmits music, enable the MSPA_MODEXMITONLY attribute in the MS_CDT data structure chan_attr field.
- The Conferencing AGC Noise Level Lower Threshold may have to be adjusted. This parameter is set at -40 dB by default and filters out any signals below this level. If the background music levels are low, the music may not be summed into the conference or it may come in and out. In this case, the AGC Noise Level Lower Threshold should be reduced by adding “setparm=0x3b1f, <value>” to the [0x3b] section of the CONFIG file and re-downloading the board. Further information about this parameter is given below.

CSUMS_AGC_low_threshold (AGC Noise Level Lower Threshold)

Number: 0x3B1F

Description: The **CSUMS_AGC_low_threshold** parameter defines the upper threshold for noise level estimates. Any signal above this threshold will be considered speech. Thus, this threshold should be set quite high in order to let the AGC algorithm determine when there are voiced and unvoiced periods. The parameter is given in terms of the average level.

CSUMS_AGC_low_threshold is defined as: $10(\text{output level in dB})/20 * 2^{23}$. Multiplying by 2^{23} converts the value into a linear 24-bit value that accommodates the 24-bit DSPs used on Dialogic® DM3 Boards.

Values: 0x0020C5 to 0x0732AE (-60 dB to -25 dB)

Default Value: 0x0147AE (-40 dB)

Guidelines: It is recommended that the value be set in the range of -60 dB to -40 dB. Do not exceed the AGC high threshold which is set to -34.6 dB in the current DM3 system.

Here is a sample calculation to get a hexadecimal value of **CSUMS_AGC_low_threshold** for a noise threshold level of -50 dB_{avg}:

$$10^{(-50/20)} * 2^{23} = 0x00679F$$

1.66.3 Documentation

The online bookshelf provided with Dialogic® System Release 6.1 for Linux contains information about all system release features including features for application development, configuration, administration, and diagnostics.

For detailed information about configuring Dialogic® DMV4800BC Boards, see the following document:

- *Dialogic® DM3 Architecture Products on Linux Configuration Guide*

For information about the Dialogic® Audio Conferencing (DCB) API, see the following documents:

- *Dialogic® Audio Conferencing API Programming Guide*
- *Dialogic® Audio Conferencing API Library Reference*

Note: The online bookshelf has not been updated for this feature, so the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* does not currently include information about media load ML10C.

1.67 Dlgsnapshot Tool Autodump Feature Now Disabled by Default

With the Service Update, the autodump feature of the dlgsnapshot tool is now disabled by default. However, you still have the option of enabling the autodump feature if you want to use it. In the default state, no board or DSP will be automatically reset as a result of a DSP failure. Previously, the entire board would be brought down if a single DSP failed. For more information about the dlgsnapshot tool, refer to the *Dialogic® System Software Diagnostics Guide*.

Release Issues

The table below lists issues that can affect the hardware and software supported in Dialogic® System Release 6.1 for Linux. 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 (i.e., issues relating to combining different Dialogic® products in the same system) and compatibility issues (i.e., issues that affect the use of Dialogic® products in with third-party software or hardware). 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 will not 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 Tracking tool at <http://membersresource.dialogic.com/defects/>.

Note that when you select this link, you will be asked to either LOGIN or JOIN.

PTR No.

Number from problem tracking system used prior to March 27, 2006. For customer convenience, both the PTR number and the corresponding defect number are shown. For issues reported after March 27, 2006, this column contains "--" and only the defect number is used to track the issue.

SU No.

For defects that were resolved in a Service Update, indicates the Service Update number. 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 issue relates, typically one of the following:

- A system-level component; for example, Host Admin
- A hardware product; for example, Dialogic® DM/V Boards
- A software product; for example, the Dialogic® Global Call library

Description

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

The following table lists all issues that relate to this release, sorted by Issue Type. For other sort orders, use the following links:

- [View issues sorted by Service Update Number](#)
- [View issues sorted by Product or Component](#)
- [View issues sorted by Defect Number](#)

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Known	IPY00006147	35635		Conferencing	When running with a combination of analog and digital boards, there is a degradation in voice quality when an analog channel is added to a digital conference.
Known	IPY00080895	--	317	Configuration	<p>Several RTF errors from <i>digclockdaemon</i> may be observed when configuring a system with JCT/Springware boards only. For example, the following may be observed in RTF:</p> <pre>ConfigTDMBus(): FATAL CONFIGURATION ERROR</pre> <p>Although these errors are harmless and can be ignored, a workaround is provided.</p> <p>Workaround: Follow the procedures in the <i>Dialogic® Springware Architecture Products on Linux Configuration Guide</i> to start the configuration utility (<i>config.sh</i>) in order to configure the TDM Bus. From the Advanced TDM Bus Settings screen, set the clocking daemon to PASSIVE or DISABLED when using JCT/Springware only boards.</p>
Known	IPY00006538	34708		D/300JCT-E1 Boards	When using CTR4 E1 protocol, T1 4ESS and 5ESS protocols on Dialogic® D/300JCT-E1 Boards, the user side does not produce a “disconnected” event when the remote side does not answer. The network side does.
Known	IPY00006303	35542		DI0408LSAR2 Boards	Dialogic® DI0408LSAR2 Board with ML4 reports 100% tone/digit generation failures when running with other non-DI0408LSAR2 Boards in the system and exportable resources are routed off the DI0408LSAR2 to the other boards. Tone/digit generation will work on ML4 if only DI0408LSAR2 Boards are in the system or if resources are not routed off the DI0408LSAR2.
Known	IPY00032251	36104		Diagnostics	its_sysinfo is referring to fwlver but unable to find it. An error message is displayed, but the failure is not reported in the sysinfo.log file.
Known	IPY00006565	34598		Diagnostics	The ISDNtrace tool consumes a portion of the CPU for each trunk that is logged and may generate an exception if CTRL-C is used to exit the tool. It may also generate an exception when invalid parameter or logical ID is entered in the command.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Known	IPY00006589	36394		DM/IP Boards	If a client is in other IP-subnetwork when using Dialogic® DM/IP Boards, the board can't deliver the RTP stream to the client. Also, the board can't ping the IP address from the remote client in other IP-subnetwork.
Known	IPY00006560	36048		DM/IP Boards	The setup and retrieval of progress message along with any signal IE is not functional.
Known	IPY00006324	36564		DM/IP Boards	When running call progress PC Fax CED, it failed with the following message: Invalid Termination reason: Expected 18, Received 10 over G.711 coder.
Known	IPY00006359	36659		DM3 Network	While 2-board NFAS is configured, dropping call from makecall side in initiated state is causing firmware timeout.
Known	IPY00079022	--		DM3 Voice	In the DV_TPT data structure, the TF_SETINIT flag for DX_MAXSIL termination condition has no effect on standard records (dx_reciottdata() family) with Dialogic® DM3 Boards, and will be ignored.
Known	IPY00029916	36581		DMN160TEC Boards	When adding Dialogic® DMT160TEC Board on Advantech 3041/3389 with Pigeon Point Systems (PPS) installed, it fails when there are no other boards in the system. Workaround: Must have at least one board installed. It will not work with no boards installed.
Known	IPY00029918	36818		DMV160LP Boards	When the digits coming in from analog network into one of the Dialogic® DMV160LP front-end devices are routed over to another DMV160LP front-end device and out to the analog network again, several digits failed to be detected.
Known	IPY00005947	36714		DMV160LP Boards	With the Dialogic® DMV160LP Board, DM3StdErr shows the following error message: Brd0 CP1: qShramDataWrite: No Node.
Known				DMV4800BC Boards	Heavy simultaneous plays or records using the 8kHz 16-bit linear coder (128 kbps) above ~ 400 channels on a Dialogic® DMV4800BC Board with ML1B might cause quality degradation on the audio.
Known				DMV4800BC Boards	Applications requiring low latency should not use MAXTIME as a termination condition when running above ~ 420 channels on a Dialogic® DMV4800BC Board with ML1B. MAXDTMF is the recommended termination condition at higher densities.
Known				DMV600BTEC Boards	On the Trunk Configuration - Specify Media Load selection screen, it lists MLFN. MLFN for 60 channel fax is not supported on the Dialogic® DMV600BTEC Board. Continue to use UL1.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Known	IPY00006271	36322		Host Admin	System intermittently receives “corrupted double-linked” message on Dialogic® IPT10000C Boards during download.
Known	IPY00032253	35829		Host Library	When running Global Call IP applications, there may be a core dump during exit. This core dump does not affect or impact the runtime operation.
Known	IPY00010766	36131		Host Library	When running GC PDK call progress tests, unexpected GCEV_BLOCKED and GCEV_UNBLOCKED events get reported.
Known	IPY00006633	36697		Host Library	A segmentation fault occurs when RTF trace is enabled for ISDN module.
Known	IPY00006823	35851		Host Runtime Libraries	The firmware crashes when CAS_Seize is similar to the wink signal.
Known	IPY00010906	36294		IPT Boards	The iptfwupdate tool fails to burn new firmware to Dialogic® IPT Boards. The failure message states “255 board does not recognize firmware.”
Known				IPT Boards	When using Dialogic® IPT Boards, if notification of all GCAMS alarms is enabled by default, some Real Time Control Protocol (RTCP) and Real Time Protocol (RTP) alarms may be generated. These alarms can be ignored.
Known				IPT Boards	When using RFC2833, G.723 coder reports higher number of missing digit/tone failures. Most of the failures are with 1 frame per packet.
Known				IPT Boards	G.729 coder reports higher number of missing digit/tone failures with VAD on.
Known				IPT Boards	H.323 with ‘Alphanumeric’ DTMF transfer mode reports higher number of missing digit/tone failures.
Known				Red Hat Linux	When running Red Hat 4.0 Update 2 Pre-emption kernel, a panic is seen occasionally during application shutdown.
Known	IPY00010862	34335		Springware Fax	Multipage fax ASCII send tests fail intermittently on Dialogic® D/600JCT Boards. Workaround: Resend the fax if a transmission fails.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Known (permanent)	IPY00030895	36381		2.6 Kernel	<p>When running 2.6 kernel on ZT5085 chassis with 4 GB SBC and Dialogic® CompactPCI Boards, there is slow OS response. The boot time exceeds 30 minutes and system is sluggish upon login.</p> <p>Workaround:</p> <p>Solution 1 - Hiding the top 96 MB of RAM from the OS prevents access problem with pages. This can be done by adding mem=4000M in boot loader file for the system. For example:</p> <p>GRUB boot loader (RHEL4: /boot/grub/grub.conf -- SLES9: /boot/grub/menu.lst):</p> <pre>kernel /vmlinuz <running kernel> . . . quiet mem=4000M</pre> <p>LILO boot loader (/etc/lilo.conf):</p> <pre>image=/boot/vmlinuz <running kernel> . . . append=". . . mem=4000M</pre> <p>Note: For LILO boot loader, after editing the file, /sbin/lilo must be executed.</p> <p>Upon reboot, the system with Dialogic® CompactPCI Boards will respond at normal speeds.</p> <p>Solution 2 - Disable 3rd level pagetable allocation from high memory feature from the 2.6 kernel (CONFIG_HIGHPT). Refer to the Linux distribution on how to rebuild the Linux kernel.</p>
Known (permanent)	IPY00010432	35500		D/300JCT and D/600JCT Boards	<p>The DSPs on the Dialogic® D/300JCT and Dialogic® D/600JCT Boards do not have enough MIPS to perform FSK processing and GTD digit processing simultaneously on the same DSP.</p> <p>Workaround: Perform FSK and GTD digit processing on different DSPs. For example, use channels 1-8 for FSK and channels 9-16 for GTD digit processing.</p>
Known (permanent)	IPY00009998	33825		D/480JCT and D/600JCT Boards	<p>For Dialogic® D/480JCT or Dialogic® D/600JCT Boards, the sender should not transmit the fax at a rate greater than 9600 bps. The maximum receive rate is 9600 bps.</p> <p>Workaround: Set the maximum send rate to match the maximum receive rate (9600 bps).</p>
Known (permanent)	IPY00030640	23783		DM/IP Boards	<p>There is no interoperability between Dialogic® DM/IP Boards and the Siemens IP phone when using the Embedded Stack. When working in Fast Start Mode and DM/IP is the originator, DM/IP sends facility with reason start H.245 which the Siemens IP phone does not support.</p> <p>Workaround: Work in Slow Start Mode.</p>

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Known (permanent)	IPY00010209	36102		DM/IP Boards	Conference density limitations to 12 conference resources on Dialogic® DM/IP Dual Span Boards. Note: This does not apply to Single Span or Resource DM/IP Boards, which always support the full 30 conference resources. Higher conference density can be achieved depending on the number of IP devices that are open and number of voice devices that are connected at a given time. Please see the Table , “Conference Density Limitations on Dialogic® DM/IP Dual Span Boards”, on page 157 for further details on what is supported.
Known (permanent)	IPY00009888	34090		DM/V Boards	When MF digits are dialed out on Dialogic® DM/V Boards, the receiving side might get an extra “c” at the beginning of the digit string.
Known (permanent)	IPY00010332	34669		DMV4800BC Boards	Simultaneous plays on high bit rate linear coders (8kHz and 11kHz with 16-bit) may experience silence on the audio as a result of audio gaps when performing over 180 channels of simultaneous play with 8kHz or 11kHz 16-bit coder.
Known (permanent)	IPY00006136	35891		DMV4800BC Boards	Heavy plays or records using the 11kHz 16-bit linear coder (176 kbps) above ~ 240 channels on a Dialogic® DMV4800BC Board with ML1B might cause quality degradation on the audio due to silence gaps.
Known (permanent)	IPY00010884	36155		Host Admin	When two Dialogic® IPT Boards are installed, the boards both have the same logical ID reported by listboards -l2. The logical ID information is not available for IPT Boards unless Peripheral Hot Swap (PHS) is active. This is unique to the IPT Boards only.
Known (permanent)	IPY00030647	34043		Host Library	During application start and stop, some memory is not returned back to the system. Explanation: Linux kernel utilizes slab allocation method to manage memory. The slab allocator does not discard used memory but rather caches them. Things like process descriptors, file descriptors, etc. are cached. After an application completes, memory may not be updated in system’s free memory. Linux kernel does this to optimize the restart of application by using the cached file descriptors, application data structures, etc. Usually the kernel memory flush or swap daemon, (which runs in the background) over time, will manage the unused cached memory. This is a general Linux system behavior.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Known (permanent)	IPY00008787	34722		Host Runtime	Streaming to board takes longer than expected on Dialogic® DM/V1200A Boards. While running 1K streaming to board tests with ml2_qs2_r2mf load on DM/V1200A Boards, the following error message was seen: qStreamGroupClose(): Unbind failed fd 34, bindhandle 238, token 0x2200ee
Known (permanent)	IPY00028577	36198		Host Runtime Libraries	When using 2.6 kernel, a segmentation fault occurs when mixing Dialogic® DM3 and Dialogic® Springware Boards in the same system. It is due to the system running out of memory. 512 MB of RAM is the minimum memory required but it is a function of the memory requirements of the application. More memory may be needed.
Known (permanent)	IPY00041217	--		PBX Call Control	Port 4 on Dialogic® D/42JCT-U Board or ports 4 and 8 on Dialogic® D/82JCT-U Board will not successfully gain carrier on an NEC PBX unless at least one other port from the same set of 4 ports is also connected to a PBX line. Workaround: Ensure that port 4 on a D/42JCT-U Board and ports 4 and 8 on a D/82JCT-U Board are never connected to the PBX without other ports in their respective 4-port range.
Known (permanent)	IPY00008617	32933		Software Requirements	Dialogic® drivers do not support Physical Address Extensions (PAE). Users using the supported Linux version need to restrict memory to 4 GB. Dialogic® drivers do not support more than 4 GB of RAM.
Known (permanent)	IPY00009973	34791		TDX_UNDERRUN	When the host is not able to supply data to board at a sufficient rate, an TDX_UNDERRUN (underrun error) event will occur at about every 1.5 seconds. The delivery of data is the problem. Workaround: Ignore the extra events messages.
Known (permanent)	IPY00037015	--		UDD	Dialogic® Diagnostics Software (UDD) reports download errors when multiple boards are installed in the same system. Workaround: When using the UDD diagnostics program to test multiple Dialogic® Springware Boards in the same system, use the configurator (CFG) to disable all boards except the one being tested.
Resolved	IPY00030665	35955	198	Call Progress	DM3StdErr reports errors while running GC call progress.
Resolved	IPY00028444	35763	189	Call Progress	The ml2_qsa_5ESS protocol does not understand call progress message when Location field is set to 1010 on outgoing call. The firmware does not properly handle the call setup when the Location field is set to 1010.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00028338	34336	171	Call Progress Analysis	The call progress analysis comes back with false cadence connects.
Resolved	IPY00036423	--	245	Conferencing	Sometimes a noise is generated when a party leaves a conference; the noise disappears when a party is added to the conference.
Resolved	IPY00028633	35748	245	Conferencing	Sometimes a noise is generated when a party leaves a conference; the noise disappears when a party is added to the conference.
Resolved	IPY00009499	33932	245	Conferencing	A loud scratch/click sound occurs when entering a conference when 1-2 parties are already in the conference.
Resolved	IPY00007470	32437	245	Conferencing	A sharp noise occurs when changing conference resource mode to MSPA_MODERECVONLY.
Resolved	IPY00006707	33803	245	Conferencing	Sometimes a noise is generated when a party leaves a conference; the noise disappears when a party is added to the conference.
Resolved	IPY00039512	--	278	Configuration	DM3 configuration files for Dialogic® DMV300BTEPE Boards using UL1 were not being generated properly. There was an incorrect setting for echo cancellation, which caused "ticks piling up" and "missed task deadline" messages in DebugAngel.
Resolved	IPY00038215	--	257	Configuration	The dcb_dsprescount() function returns an incorrect value. It returns double the resources.
Resolved	IPY00037612	--	248	Configuration	When running config.sh during the initial configuration process, if you select option 3 to configure Dialogic® IPT Boards, the error message "dlgOrbcus_start: command not found" is returned.
Resolved	IPY00036955	--	251	Configuration	When multiple Dialogic® D/82JCT-U Boards are installed in the same system, they cannot be configured. The config.sh board summary screen shows both boards with "Thumb Wheel" number 0, but D/82 Boards do not have configurable thumbwheel switches.
Resolved	IPY00036920	--	247	Configuration	When clocking mode is set to PASSIVE, some Dialogic® DM3 Boards are not put on the CT Bus.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00036877	--	247	Configuration	<p>In a mixed system with Dialogic® DM3 and Dialogic® Springware Boards, a clocking configuration with a DM3 Board as the Primary Clock Master and a Springware Board as the Reference Master does not work.</p> <p>Note: Mixed DM3 and Springware systems must not have split CT Bus Clock Master roles. To address this defect, a Warning message has been added in rtlog to alert the user of a potential CT Bus clock incongruence.</p> <p>To eliminate the potential of having unpredictable CT Bus clock settings in the specific case of a mixed system where the Clock Masters must be split between the two technologies, e.g., Clock Master DM3 and Reference Master Springware, you must disable the default ClockDaemonMode parameter (ACTIVE) and set it to PASSIVE. All clock fallback functionality is then disabled to avoid the incongruence.</p> <p>For information about disabling the clock daemon, refer to the procedures for configuring the TDM bus in the <i>Dialogic® DM3 Architecture Products on Linux Configuration Guide</i> and the <i>Dialogic® Springware Architecture Products on Linux Configuration Guide</i>.</p>
Resolved	IPY00036035	--	237	Configuration	Dialogic® DM/V1200BTPE Board fails to download E1/T1 clear channel protocol with “waiting for 0x9 message timed out” on Service Update 232.
Resolved	IPY00035860	--	237	Configuration	Dialogic® DM/IP241-1T1-100BT Board fails to download with “waiting for 0x5 message timeout” on Service Update 217.
Resolved	IPY00044257	--	294	CSP	With Dialogic® D/41JCT-LS Board, the MD_GAIN parameter for automatic gain control has effect on the recording with dx_reciottdata() , but has no effect on the recording with ec_reciottdata() .
Resolved	IPY00041983	--	279	CSP Demo	After executing the Dialogic® CSPLive demo, and then issuing the dmesg command, the output shows some kernel errors.
Resolved	IPY00034678	--	241	CSP Demo	Dialogic® CSPAuto demo fails to return TEC_STREAM event if more than one process is run per board.
Resolved	IPY00081390	--	317	D/4PCIE Boards	Multiple D/4PCIE boards in the system caused the system to freeze.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00007340	26773	--	D/82JCT-U Boards	Pressing the transfer key a second time to complete the transfer takes about 4-5 seconds to return TDX_DIAL event.
Resolved	IPY00045503	--	307	Diagnostics	When debugging a potential board assert on a system with multiple Dialogic® D/82JCT-U Boards, the its_sysinfo tool only produced the assert information for one of the boards.
Resolved	IPY00045343	--	300	Diagnostics	The pritrace utility is only installed with the RPM for Dialogic® Springware Boards, but it is supported on Dialogic® DM3 Boards as well.
Resolved	IPY00045159	--	300	Diagnostics	When using the PSTN Diagnostics tool (pstndiag), the following error occurs after clicking on a channel of an installed Dialogic® DM3 PSTN Board: "Error: Can't read alarms Trans (0x1616): no such element in array."
Resolved	IPY00045021	--	298	Diagnostics	The dm3post tool reports that boards have an error, with diagnostic codes 0xff 0x63, but the boards are working.
Resolved	IPY00037721	--	257	Diagnostics	Running RTFManager.sh causes an error message ": No such file or directory/usr/dialogic/" to be printed.
Resolved	IPY00037708	--	251	Diagnostics	The its_sysinfo tool, which is used to collect data including a PCI firmware dump, is not collecting a full memory dump.
Resolved	IPY00030001	36796	257	Diagnostics	ISDNtrace utility fails to start for NI2 and QSIG protocols.
Resolved	IPY00079703	--	312	Dialogic® HMP software	A TF_SETINIT setting is ignored on Dialogic® HMP software for DX_MAXSIL record termination.
Resolved	IPY00079678	--	312	Dialogic® HMP software	Echo cancellation on Dialogic® DNI boards does not function properly.
Resolved	IPY00041078	--	275	DM/IP Boards	Unknown audio or DTMF is being sent from a Dialogic® DM/IP Board at the beginning of a SIP call, which precedes the expected audio to be heard from the file played.
Resolved	IPY00028485	36093	189	DM/IP Boards	When running T.38 tests, some faxes in the log file are passing while others are failing with zero bytes.
Resolved	IPY00010721	36075	189	DM/IP Boards	When using ADDRESSTYPE_PHONE type aliases, it fails on H.323 Gatekeeper registration.
Resolved	IPY00009534	35090	--	DM/V Boards	When calling from Dialogic® JCT Boards to Dialogic® DM/V Boards, SIT is not detected.
Resolved	IPY00028463	35925	189	DM/V-A Boards	The Saturation Alarm Generation feature is not supported on Dialogic® DM/V-A Board PDK loads. This feature is still present on all Dialogic® DM/V-B and Dialogic® DMN160TEC Board loads, as well as all Dialogic® DM/V-A Board ISDN loads.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00079345	--	309	DM3 Call Control	Missing GCEV_DROPCALL event when using DPNSS with Dialogic® DM3 Board. The application made an outbound call using gc_makeCall() and the call was dropped with no answer. Then gc_DropCall() was issued after receiving GCEV_DISCONNECTED, but GCEV_DROPCALL was not received.
Resolved	IPY00079160	--	308	DM3 Call Control	When using NI2 protocol on Dialogic® DM3 Boards, gc_GetCallInfo() did not retrieve the ANI when Numbering Plan ID was "Private"; it returned with a blank ANI.
Resolved	IPY00045132	--	298	DM3 Call Control	Under heavy load on certain Dialogic® DM3 PSTN Boards, calls that are offered to channels might get dropped immediately. They moved from Answering to Disconnected state due to an overlapping of call indexes among multiple network interfaces.
Resolved	IPY00041407	--	280	DM3 Call Control	When setting up NFAS for the 4 lines on a Dialogic® DM3 T1 Board using DMS protocol. the board does not respond to a RESTART ACKNOWLEDGEMENT transmitted.
Resolved	IPY00041233	--	276	DM3 Call Control	When a call is terminated in the GCST_DETECTED state, a fake GCEV_OFFERED event should not be generated if the application has enabled the GCEV_DETECTED event.
Resolved	IPY00041209	--	280	DM3 Call Control	GCEV_UNBLOCKED event doesn't arrive for individual channels, even though GCEV_BLOCKED was delivered to individual channels, after AIS alarms occur and are then cleared.
Resolved	IPY00038494	--	267	DM3 Call Control	CP failure on Dialogic® DM3 PSTN boards triggered by specific inbound call conditions.
Resolved	IPY00037841	--	262	DM3 Call Control	When using ISDN under DM3, gc_open() fails after hot swap test. The dm3cclib cancels 2 events when closing devices (detected and offered), but only waits for one.
Resolved	IPY00037507	--	257	DM3 Call Control	Event API fails to deliver an event when the T1 is configured for CAS and the cable is unplugged.
Resolved	IPY00034857	--	257	DM3 Call Control	When performing call progress analysis via the Global Call mediadetected method, if the media detection occurs before the out-of-band CONNECT message is received, GCCT_UNKNOWN is returned as a result.
Resolved	IPY00045184	--	300	DM3 Conferencing	The dcb_dsprescount() function returns the incorrect number of resources for Dialogic® DM/IP241-1T1 Boards.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00044132	--	293	DM3 Conferencing	The dcb_unmonconf() function removes the wrong conferee (party) from the conference. Instead of removing the monitor conferee, the most recently added conferee is removed from the conference.
Resolved	IPY00041111	--	276	DM3 Conferencing	When using the ms_dsprescount() function, the function doesn't return and the application freezes.
Resolved	IPY00039068	--	271	DM3 Conferencing	The dcb_addtoconf() function returns failure, and ATDV_ERRMSGP shows the error message as "Timed out waiting for reply from firmware."
Resolved	IPY00038919	--	267	DM3 Conferencing	The ATDV_ERRMSGP() function returns "No error" when the dcb_setcde() function returned failure.
Resolved	IPY00037861	--	257	DM3 Conferencing	If one conferee goes on mute, other conference participants hear buzzing noise. Note: A documentation update to Section 6.8, [0x3b] Parameters (parameters 0x3b03 and 0x3b04) has been added in the Documentation Updates section for the Dialogic® DM3 Architecture Products on Linux Configuration Guide . Please refer to it for information relevant to this defect resolution.
Resolved	IPY00037817	--	257	DM3 Conferencing	When playing background music through the telephone set, music cuts are heard when party A speaks.
Resolved	IPY00037396	--	257	DM3 Conferencing	Static background noise trails voice in conferences with more than 6 parties.
Resolved	IPY00037373	--	257	DM3 Conferencing	In a conference with two parties, if party A keeps speaking while party B starts speaking, party B hears breaks from party A while party B is speaking.
Resolved	IPY00045442	--	301	DM3 CSP	Dialogic® DM3 Board channel hangs when failing to listen to a TDM bus time slot prior to invoking the ec_stream() function.
Resolved	IPY00044087	--	292	DM3 CSP	MAXSIL termination condition does not cause termination when using raw data mode (wDataFormat = DATA_FORMAT_RAW) with the default tp_flags (TF_MAXSIL) even if the proper conditions occur. Workaround: When DX_MAXSIL and DX_MAXNOSIL termination conditions are used, by default the silence timer starts after a dx_play() has completed. Since with raw data mode, a play is normally not used with the CSP recording session, start the silence timer immediately at the onset of ec_stream() or ec_reciottdata() by using the bit flag TF_IMMEDIATE in the tp_flags.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00043314	--	289	DM3 Drivers	Kernel panic errors occur in particular application scenario with large densities and when using the one process-per-channel application model.
Resolved	IPY00045376	--	301	DM3 DTI	After setting event masks and then trying to retrieve the masks using dt_getevtmask() , this function failed with "Unknown error" as the reason.
Resolved	IPY00043443	--	289	DM3 Fax	Fax TIFF files are received with incorrect width; half pages are received.
Resolved	IPY00043240	--	289	DM3 Fax	An exception is generated during fax call tear-down process after fx_stopch() is issued, causing the application to stop running.
Resolved	IPY00041421	--	280	DM3 Fax	Fax channels may hang when a stop is issued at the end of a send fax page.
Resolved	IPY00041079	--	280	DM3 Fax	The fx_rcvfax() function returns -1 error after the system is running for several days, and the system is not able to receive faxes.
Resolved	IPY00039661	--	273	DM3 Fax	ATFX_RESLN() sometimes returns 0, which is an invalid value. (According to the documentation, the only valid values are 98 and 196.) Note: A documentation update has been added in the Documentation Updates section for the Dialogic® Fax Software Reference . Please refer to it for information relevant to this defect resolution. There are additional return values for ATFX_RESLN() , and the values passed to fx_rcvfax() and fx_sendfax() have more options. (The defect number associated with the documentation update is IPY00040796.)
Resolved	IPY00039539	--	271	DM3 Fax	When sending multiple faxes within a single call with the T.30 EOM, the software returns an error saying the remote side is disconnected while it is not.
Resolved	IPY00039476	--	271	DM3 Fax	Stuck fax channels during inbound calls.
Resolved	IPY00038407	--	260	DM3 Fax	ATFX_RESLN() sometimes returns 0, which is an invalid value. (According to the documentation, the only valid values are 98 and 196.)
Resolved	IPY00037166	--	257	DM3 Fax	After an inbound fax call, the fax resource cannot go back to idle after fx_stopch() .
Resolved	IPY00032797	--	257	DM3 Fax	The fax sender cannot wait to receive retry of digital identification signal (DIS) message, and gets Phase E status (EFX_COMMERRTX) transmit communication error.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00010540	35853	171	DM3 Fax	Unable to send 5 page TIFF and JPEG at the same time. On Dialogic® DM3 Boards that support fax functionality, when faxing an image following a multi-page TIFF in the same fax session, it was observed that the image did not go through and the app receives a fax error event. If the image is sent after a single page TIFF or ASCII file, the entire fax session transmission goes through.
Resolved	IPY00010022	34801	--	DM3 Fax	TFX_FAXERROR while receiving JPEG fax image.
Resolved	IPY00008556	30429	--	DM3 Fax	When sending faxes over the IP via T.38 the "TO" (FC_REMOTID) field in the fax header is empty on half the channels.
Resolved	IPY00079213	--	309	DM3 Firmware	See the description for IPY00078576.
Resolved	IPY00078445	--	304	DM3 Firmware	When using Dialogic® DM3 DM/V-B Boards, DSP crashes "KillTask" were seen in the DebugAngel log during standard playback; the affected channels could not be recovered without a board reinitialization.
Resolved	IPY00045440	--	301	DM3 Firmware	Dialogic® DM/V-B Board could not detect DTMF digits when digits are on about 0 dB per frequency.
Resolved	IPY00045388	--	301	DM3 Firmware	Playing a wave file with an invalid byte count in the header caused Dialogic® DM3 firmware to crash.
Resolved	IPY00045277	--	300	DM3 Firmware	An intermittent, partial PCM data stream corruption on network interface channels on Dialogic® DM/V1200BTEP Media Boards was observed during a local loopback mode test where every channel loops back its incoming stream out to the network.
Resolved	IPY00044832	--	297	DM3 Firmware	Under certain conditions and tone templates for a dx_playtone() in asynchronous mode, the application does not receive the TDX_PLAYTONE event on the voice channel.
Resolved	IPY00043609	--	291	DM3 Firmware	When ms_listen() is called on more than 15 station interface devices of a Dialogic® DI/SI32 Switching Board, dx_playiotdata() returns TDX_ERROR and playback is not possible on the first 16 voice resources of the DI/SI32.
Resolved	IPY00043275	--	288	DM3 Firmware	Many queue_put_msg_array messages are seen in the <i>/var/log/messages</i> file. When this occurs, the system becomes unresponsive and needs power cycles to resolve the issue. This occurs when the host system ceases to process record events for extended periods of time, in seconds.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00043077	--	290	DM3 Firmware	Inbound ISDN calls made from cell phone failed to get answered on system connected to NET5 line. The calls get rejected with STATUS message with cause value of "100", which indicates invalid information element contents. The Progress Indicator IE containing Progress Description 16 and Progress Location 1 is being rejected.
Resolved	IPY00041740	--	280	DM3 Firmware	Local pool size for GTD needs to be increased for Dialogic® DMV-B Boards.
Resolved	IPY00041580	--	280	DM3 Firmware	Over time (usually past 24 hours), dialed DTMFs get corrupted. When a '5' is dialed, the digit might get repeated in fragmented stutter and/or followed by an '8' even though never specified in the dialstring.
Resolved	IPY00009451	33530	204	DM3 Firmware	After running for several weeks, half or all voice channels on a Dialogic® DM/V2400A Board are blocked. They stayed in stopped state and could not go back to idle state.
Resolved	IPY00043545	--	290	DM3 Host Runtime Library	Call to ipm_Close() hangs with Dialogic® DI/0408-LS Switching Board if there are no additional DM3 voice resources in the system.
Resolved	IPY00043430	--	289	DM3 Host Runtime Library	After performing a peripheral hot swap of Dialogic® DM/IP Boards, and boards are successfully restarted, gc_OpenEx() fails to open on-board IPML devices, returning error 0x44, Invalid Parameter.
Resolved	IPY00038998	--	267	DM3 Host Runtime Library	Bipolar violation alarms are properly detected by the board but are not being reported programmatically via GCAMS.
Resolved	IPY00034559	--	239	DM3 IP	Assert in the RADVISION stack.
Resolved	IPY00028657	36577	198	DM3 IP	If ipm_stop() is used in an IPML application, Dialogic® DM3 IP Board cannot send out the RTP packet at all, after running IPML application the second time.
Resolved	IPY00028319	35794	171	DM3 IP	An invalid value is not returned when an out of range payload type is issued. When this occurs, ipm_StartMedia() or Global Call failures are seen. Make sure that the application using payload type for RFC2833 is always between 96 and 127.
Resolved	IPY00007718	30630	--	DM3 IP	When using G.711, the application should use the same frame size for coders as the one used by the remote side (if known).
Resolved	IPY00007640	30390	--	DM3 IP	Implementation of asymmetric coders for transmit and receive should not be used. By setting the same coder for transmit and receive in the application, asymmetric coder negotiation could be avoided.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00007555	29164	--	DM3 IP	The error message (RTP Recv: in media_RTPUnpack() SSRC failed) may occasionally be seen in DM3StdErr when running DM/IP configurations.
Resolved	IPY00010242	34320	--	DM3 Network	When SetAlarmFlow is set to have the app receive alarm at Blocking and NonBlocking, app only receives alarm during Blocking.
Resolved	IPY00008084	32160	--	DM3 Network	The progress message is incorrectly sent before the proceeding message.
Resolved	IPY00007908	28288	--	DM3 Network	gc_GetSigInfo() fails sometimes on OFFERED event.
Resolved	IPY00007844	27539	--	DM3 Network	If a call is received on Q.931 where there is no channel ID in the setup message, the call is rejected.
Resolved	IPY00007686	31991	198	DM3 Network	When configuring inter-board NFAS where the Primary and Secondary D channel are on separate boards, NFAS trunks on the board with the Secondary D channel cannot make or accept calls. However, NFAS trunks on the Primary D channel board (intra-board NFAS) are not affected and calls can successfully be placed. If the Data Link on the Primary D channel is taken down, the Standby D channel does not successfully take over and now NFAS trunks on the both boards cannot make or accept calls.
Resolved	IPY00007526	25549	--	DM3 Network	During initialization sequence a DMS100 switch sets each bearer channel out of service and then issues a D-channel reset.
Resolved	IPY00007370	27563	--	DM3 Network	For DMS100 protocol after restart message, inbound calls are rejected with cause code 44 channel busy.
Resolved	IPY00007288	27764	--	DM3 Network	Outbound calls fail when the ALERTING message contains a Non-Locking Shift IE.
Resolved	IPY00007234	23614	--	DM3 Network	When a trunk receives AIS, the LineAdmin utility only displays the green and red LED lights, but not the yellow.
Resolved	IPY00007216	26043	--	DM3 Network	The NI2 firmware identifies itself as 4ESS when using a utility like ISDNTRACE.
Resolved	IPY00038533	--	260	DM3 Runtime Libraries	An internal parameter is not decremented correctly when a process exits, causing failures in opening devices.
Resolved	IPY00007463	29364	--	DM3 Tools	Running ISDNtrace uses up all available CPU cycles.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00079212	--	309	DM3 Voice	After dx_getdig() was called and returned, and dx_clrdigbuf() was called and returned, ATDX_BUFDIGS() still reported that there were digits in the buffer; it should have returned 0. The problem occurred when dx_getdig() terminated due to IDDTIME (inter digit delay).
Resolved	IPY00079206	--	308	DM3 Voice	DSPs were not able to keep up while running an application that is media intensive (plays/records) with 64 Kbps G.711 audio on a Dialogic® DM/V4800BC Board using <i>ml1b_cpciresb.pcd</i> configuration; at some point I/O operations started to fail simultaneously on multiple voice channels, or otherwise they would get stuck in the middle of the operation, e.g., dx_playiottdata() or dx_reciottdata() API call.
Resolved	IPY00079095	--	308	DM3 Voice	Under certain race conditions, a dx_playiottdata() caused an internal thread deadlock in the Voice library, leading to an application core dump.
Resolved	IPY00045293	--	301	DM3 Voice	Dialogic® DM3 Board channel hangs when failing to listen to a TDM bus time slot prior to invoking a record operation (dx_reciottdata() or similar voice recording function).
Resolved	IPY00044932	--	297	DM3 Voice	A voice stuck channel occurred during a playback operation while running an application that is media intensive (plays/records) due to an internal race condition. No TDX_PLAY event is returned to the application after dx_playiottdata() is issued.
Resolved	IPY00044811	--	297	DM3 Voice	A voice stuck channel occurred during a playback operation while running an application that is media intensive (plays/records) due to an internal race condition. No TDX_PLAY event is returned to the application after dx_playiottdata() is issued.
Resolved	IPY00044614	--	297	DM3 Voice	A potential voice stuck channel condition that could occur on Dialogic® DM/V-B Boards during a record (dx_reciottdata()) operation was caused by a race condition in the record data stream handling when a stop channel (dx_stopch()) is issued to terminate the operation.
Resolved	IPY00044561	--	297	DM3 Voice	A voice stuck channel condition occurred, caused by the unhandling of tone creation failures during the setting of tone termination conditions for a playback (dx_playiottdata()) or record (dx_reciottdata()) operation. In asynchronous mode the TDX_PLAY or TDX_RECORD events, respectively, would never be received.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00044432	--	294	DM3 Voice	A firmware stream corruption was seen during the execution of a very intensive media application under specific conditions; this would cause a single voice channel not to return a TDX_RECORD event after dx_reciottdata() function had been issued, causing the channel to be stuck.
Resolved	IPY00044363	--	294	DM3 Voice	A firmware DSP crash occurred during the running of an application that is media intensive (plays/records) due to an internal race condition; at some point plays start to fail simultaneously on multiple voice channels with RTF logs showing "Std_MsgError in PlayerStartingPlayer : Error Message 0x128" on groups of voice channels when attempting to execute a dx_play() API call; similar results would occur on any other I/O voice function.
Resolved	IPY00044185	--	293	DM3 Voice	The dx_setevtmask() function with DM_DIGOFF only disabled events if DM_DIGITS was used first, i.e., they had to be first enabled; thus default user-defined tones could not be disabled.
Resolved	IPY00043907	--	292	DM3 Voice	A dx_stopch() issued during a narrow window of time from a dx_play() start caused the voice channel to get stuck in this state and never return a termination event TDX_PLAY.
Resolved	IPY00043818	--	292	DM3 Voice	Receipt of TDX_RECORD events is delayed after calling dx_stopch() on Dialogic® DM/V4800BC Boards configured with the ML1B load at full density and under high load.
Resolved	IPY00043701	--	291	DM3 Voice	No TDX_PLAY event is ever returned from dx_stopch() when the stop is issued during a playback operation and the time of the request coincides with some internal playback setup states timing window on the affected channel.
Resolved	IPY00043432	--	289	DM3 Voice	The dx_playiottdata() function returns TDX_ERROR when attempting to play an empty wave file on Dialogic® DM3 Boards.
Resolved	IPY00042860	--	292	DM3 Voice	Receipt of TDX_RECORD events is delayed after calling dx_stopch() on Dialogic® DM/V4800BC Boards configured with the ML1B load at full density and under high load.
Resolved	IPY00042845	--	287	DM3 Voice	Single channel play failures occur when running an application that handles a lot of plays/records concurrently and repeatedly in a live environment. At some point, a single play fails to return a TDX_PLAY event, and even calling dx_stopch() will leave that channel in a stuck state.
Resolved	IPY00042828	--	287	DM3 Voice	Indexed wave file doesn't play as per the given indexes to dx_playiottdata() ; instead, the complete file is played.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00041129	--	275	DM3 Voice	When short buffers (below 4k) are configured with dx_setchxfercnt() and used with a UIO play, the application's callback function requested additional input beyond the configured size. Because of this, data was overwritten and channel output was distorted.
Resolved	IPY00041088	--	276	DM3 Voice	Failures occur when trying to play voice files from multiple sources, e.g., one from memory and one from a file by using dx_playiottdata() .
Resolved	IPY00040832	--	273	DM3 Voice	TEC_STREAM event is not returned to the application when ec_stopch() is called after dx_unlisten() is performed on that voice channel.
Resolved	IPY00040685	--	273	DM3 Voice	ATDX_TRCOUNT() returns the wrong value when playing a GSM 6.10 WAVE file on Dialogic® DM3 Boards.
Resolved	IPY00039586	--	271	DM3 Voice	ERROR_BROKEN_PIPE error internal message is reported in RTF logs during a streaming to board play.
Resolved	IPY00039412	--	271	DM3 Voice	TDX_PLAY is not generated to the application during streaming to board play; dx_GetStreamInfo() is not returning correct information.
Resolved	IPY00039032	--	271	DM3 Voice	DM3 Voice resources don't go to idle state after dx_stopch() function.
Resolved	IPY00038991	--	267	DM3 Voice	Previously existing user-defined tones are still being detected after deletion (i.e., call dx_deltone()) on the same channel in which a new set of different user-defined tones have been created.
Resolved	IPY00038981	--	268	DM3 Voice	TDX_PLAY is not generated to the application during streaming to board play; dx_GetStreamInfo() is not returning correct information.
Resolved	IPY00038611	--	262	DM3 Voice	When using the dx_playtone() function with TONEON or TONEOFF as the terminating condition, when the TONEON or TONEOFF event occurs, the program gets a TDX_ERROR event instead of TDX_PLAYTONE event.
Resolved	IPY00037183	--	257	DM3 Voice	When recording WAV 176 bps file (11 KHz, 16 bits per sample), dx_mreciottdata() stops prematurely with EOD before recording all bytes specified in io_length field of DX_IOTT structure, if this field is set to some large value (in this case, 26 Mb). Other formats, such as 64 kbs PCM MuLaw, ALaw, Linear, and ADPCM did not exhibit this problem.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00036865	--	248	DM3 Voice	If a user attempts to do a play forever (specifying <code>io_length = -1</code>) with UIO plays on Dialogic® DM3 Boards, there is still a hard upper limit on the number of bytes that can be played, which is approximately equal to 2.147 GB (~2 to the 31 bytes).
Resolved	IPY00036861	--	245	DM3 Voice	When attempting to run transaction recordings under rapid succession, sometimes the internal CT Bus routing fails and the record returns with a <code>TDX_ERROR</code> event with the result "Switching Handler is not Present."
Resolved	IPY00030894	34931	--	DM3 Voice	There's a higher percentage of record failures when all 480 channels are running the 11kHz 16-bit coder on ML1B. When 480 channels are run with a mix of various coders, the issue does not appear.
Resolved	IPY00030654	34969	--	DM3 Voice	If <code>dx_play()</code> is issued when using UIO, the buffer size is overwhelmed.
Resolved	IPY00009959	34011	--	DM3 Voice	<code>ATDX_BUFDIGS</code> is not implemented for Dialogic® DM3 Boards.
Resolved	IPY00009619	34010	--	DM3 Voice	Function <code>ATDX_CHNAMES()</code> returns NULL pointer for Dialogic® DM3 Boards.
Resolved	IPY00008426	33366	--	DM3 Voice	Using the <code>io_length</code> to specify the number of bytes to record, the <code>dx_rec()</code> function records more bytes than expected.
Resolved	IPY00006647	36598	213	DMV1200BTEC Boards	Running <code>Simul_Inx</code> back-to-back on a Dialogic® DMV1200BTEC Board is causing <code>mem_free()</code> failures in the <i>debugangel.log</i> .
Resolved	IPY00028556	35819	189	DMV1200BTEP Boards	The ML5B media load on Dialogic® DMV1200BTEP Boards supports only 6 fax channels, although it is documented as supporting 120 transaction records in the <i>Dialogic® DM3 Architecture Products on Linux Configuration Guide</i> .
Resolved	IPY00009694	33925	--	DMV1200BTEP Boards	Stopping a simple play application using <code>kill -9</code> causes the Dialogic® DMV1200BTEP Board to crash.
Resolved	IPY00028474	35874	189	DMV3600BP Boards	The UL2 media load on Dialogic® DMV3600BP Boards supports only 34 concurrent transaction records, although it is documented as supporting 12 fax channels in the <i>Dialogic® DM3 Architecture Products on Linux Configuration Guide</i> . Workaround: Use the ML5BC or UL1 media load, which supports full density transaction records.
Resolved	IPY00028581	35821	171	DMV4800BC Boards	The ML2 media load on Dialogic® DMV4800BC Boards should be named ML2C since it provides CT Streaming of CSP data.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00009223	35046	--	DMV600BTEC Boards	The Blue LED light on the faceplate of Dialogic® DMV600BTEC Boards does not turn on when a removebrd is issued. In addition, clicking the red snap button does not turn on the Blue LED light.
Resolved	IPY00079399	--	311	Fax	When the QFC3 or QFC2 tries to destroy the handle of a closed stream, the operation fails and causes a KILLTASK.
Resolved	IPY00036280	--	243	Fax	When a Dialogic® VFX/41JCT-LS Board is receiving fax when the line quality is not good, sometimes the calls are terminated by error with ESTAT 193.
Resolved	IPY00035660	--	234	Fax	Newline character in ASCII fax corrupts received tiff file.
Resolved	IPY00034105	--	241	Fax	Dialogic® VFX41JCT-LS Board channel would become unable to send/receive fax after particular fax call scenario occurs.
Resolved	IPY00033472	--	243	Fax	Specifying multiple DTMF detection methods or IP_DTMF_TYPE_RFC_2833 prevents fax CED detection.
Resolved	IPY00031534	--	241	Fax	When sending a fax, the Dialogic® VFX/41JCT-LS Board cannot establish phase B with some particular fax machine.
Resolved	IPY00011037	36677	239	Fax	Inbound fax call fails when a previous call on the same device is dropped and media devices are disconnected using gc_SetUserInfo() .
Resolved	IPY00034036	--	239	Gatekeeper Registration	The gatekeeper may not respond to keep-alive registration within 2 seconds, causing GCEV_TASKFAIL at application level.
Resolved	IPY00010726	36125	189	Gateway_R4 Demo	The Gateway_R4 demo is using old libdata initialization, causing gc_Start() to fail when using GC_H3R_LIB cclib.
Resolved	IPY00034841	--	239	Global Call	While closing the H.323 channels, some of the H.323 channels may not close properly, causing the subsequent events to be directed to incorrect devices.
Resolved	IPY00030623	29291	--	Global Call Demo	After a period of time, channels seem to stop detecting and responding to bit states/transitions.
Resolved	IPY00007364	28685	--	Global Call Demo	The Global Call demo fails when using analog protocol.
Resolved	IPY00081061	--	317	Global Call IP (SIP)	The SIP stack does not negotiate correctly when VAD is specified as DON'TCARE on the local side. VAD-specific coders in an incoming SDP are not accepted.
Resolved	IPY00079716	--	315	Global Call IP (SIP)	Incoming calls on an IPT board are rejected with a GCEV_TASKFAIL (IPERR_INVALID_PHONE_NUMBER) event.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00079691	--	315	Global Call IP (SIP)	IPT responds to a reINVITE but loses RFC2833 indication on 200_OK.
Resolved	IPY00079668	--	315	Global Call IP (SIP)	On a reINVITE on an IPT board, the IP_CAPABILITY structure reported G729A instead of G729AB.
Resolved	IPY00079651	--	315	Global Call IP (SIP)	The gc_AcceptModifyCall() function fails with IPERR_BAD_PARAM.
Resolved	IPY00079648	--	315	Global Call IP (SIP)	On a reINVITE on an IPT board, the IP_CAPABILITY structure reported at GCEV_REQ_MODIFYCALL has an incorrect "audio.frames_per_packet" field value when using a low bit rate codec (G729A/AB/G723).
Resolved	IPY00079393	--	311	Global Call IP (SIP)	The SIP Allow header is omitted in response messages to inbound calls.
Resolved	IPY00079108	--	315	Global Call IP (SIP)	IPPARM_OFFERED_FASTSTART_CODER/GCSET_CHAN_CAPABILITY reporting for SIP G723.1 is always chosen at the default bit rate of 6.3k if remote side does not specify the bit rate.
Resolved	IPY00042168	--	283	Global Call Protocols	When running United States T1 Bidirectional protocol (pdk_us_mf_io), blind transfer failure scenario is not handled properly. In a failing scenario where gc_BlindTransfer() yields a GCEV_DISCONNECTED event to the application, a subsequent gc_DropCall() produced no GCEV_DROPCALL event as it should have.
Resolved	IPY00041855	--	283	Global Call Protocols	Call could not be completed because the Mexico R2 protocol failed to send additional DNIS digits.
Resolved	IPY00028383	35321	--	Global Call Protocols	Busy tones are detected as "no ringback" in call progress analysis when using the dx_dial() method in Global Call application.
Resolved	IPY00028378	34586	--	Global Call Protocols	For inbound call, channel is blocked after the remote caller hangs up before sending DNIS, when using pdk_hk_dtmf_io.cdp.
Resolved	IPY00010746	35042	--	Global Call Protocols	When using the pdk_us_mf_io protocol, if CDP_OUT_Send_Alerting_After_Dialing = 1 and CPA is disabled, the user expects to get the GCEV_ALERTING event right after dialing. However, if the remote side answers the call too quickly, the GCEV_CONNECTED event is returned and the GCEV_ALERTING event never comes in.
Resolved	IPY00010621	34537	--	Global Call Protocols	When using the pdk_us_mf_io protocol in the Feature Group D configuration, ANI is missing the last digit when ANI is not terminated with the expected ST digit.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00010372	35035	--	Global Call Protocols	After sending CAS_HOOKFLASH, there should be some delay before sending DTMF in pdk_sw_e1_necls_io protocol, if CDP_WaitDialToneEnabled = 0 (i.e., do not wait for dialtone).
Resolved	IPY00010223	34985	--	Global Call Protocols	pdk_sw_e1_ermx_io.cdp can only accept one ringing signal (the internal ringing or the external ringing but not both). Defining CAS_RING_APPLIED (0001 -> 0xxx) solves the detection of the two ringing signals but causes problems with outgoing calls.
Resolved	IPY00010129	34274	--	Global Call Protocols	Global Call does not provide a way to disable DISCONNECT TONE SUPERVISION with pdk_na_an_io.cdp.
Resolved	IPY00010035	35159	--	Global Call Protocols	Under certain conditions when a gc_MakeCall() attempt times out, it incorrectly displays the result message as NORMAL CLEARING instead of timeout.
Resolved	IPY00010004	34685	--	Global Call Protocols	When using the pdk_us_mf_io protocol in the Feature Group D configuration, the protocol does not send a Disconnect signal when it times out waiting for DNIS and ANI. This occurs when the remote side is configured as Feature Group B and makes a call.
Resolved	IPY00009837	35049	--	Global Call Protocols	There seems to be a hard-coded 30-second timeout on a Make Call when the call is made in Alerting mode, which will terminate the call if no one picks up the phone. The expected behavior is that the call will not be dropped automatically, so the phone will ring forever if no one picks up. This occurs on T1 CAS lines.
Resolved	IPY00009409	34663	--	Global Call Protocols	When using FXS protocol and calling a busy station using supervised transfer, you get a disconnect event for both the consultation CRN and transferred CRN.
Resolved	IPY00008220	34972	--	Global Call Protocols	When using the pdk_us_mf_io protocol, digits from the previous call are returned in ANI.
Resolved	IPY00007327	30233	--	Global Call Protocols	With the pdk_mx_r2_io protocol, if the E1 cable is disconnected and reconnected, the application does not receive all the GCEV_UNBLOCKED events.
Resolved	IPY00006809	34543	--	Global Call Protocols	When CDP_IN_DNIS_ST_Needed = 0, the pdk_e1_cas_io protocol should not issue timed-out error while waiting for DNIS.
Resolved	IPY00006804	34319	--	Global Call Protocols	If a board is configured using pdk_us_ls_fxs_io.cdp file and a call is abandoned after the first ring, the application is not receiving the GCEV_DISCONNECTED event that is expected.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00006771	34329	--	Global Call Protocols	Using Belgium R2 protocol, when configured in DTMF/MF mode, in the OFFERED state the gc_ResetLineDev() function does not behave properly.
Resolved	IPY00006762	34664	--	Global Call Protocols	When using E1 line side protocol and calling a busy station using supervised transfer, you get a disconnect event for both the consultation CRN and transferred CRN.
Resolved	IPY00006748	34587	--	Global Call Protocols	The PDK E1 CAS protocol cannot be downloaded on Dialogic® DM3 Boards, and Dialogic® Springware Board channels cannot be opened when using this protocol.
Resolved	IPY00006735	34344	--	Global Call Protocols	On Dialogic® DM3 Boards, when dialtone is enabled on Belgium R2 protocol, if the first DTMF/MF digit of DNIS sent is 1 then the DNIS digits received at the inbound side are not the same as sent by the outbound side.
Resolved	IPY00037002	--	251	Host Admin	When running dlstop to terminate Dialogic® services, it completes successfully according to output written to the screen. However, after it completes, you can find a core file from dlgsysclockdaemon, which was generated from the shutdown of services in the '/' directory.
Resolved	IPY00032254	36452	204	Host Admin	dlstart creates additional dlgsysmonitorserver and dlgsysclockserver processes when running more than once without dlstop in between.
Resolved	IPY00032249	36682	205	Host Admin	The Dialogic® VFX/41JCT-LS Board fails to download on with the following error message: dxxxB1: Failed open in scsstdtdxag.c: No such file or directory.
Resolved	IPY00032055	36481	204	Host Admin	Five elements in the DLGHWINF MIB are of type integer but should be either Gauge32 or Unsigned32. Errors are returned by UCD-SNMP when they are queried. Some values are overflowing to negative numbers.
Resolved	IPY00031592	36725	205	Host Admin	When running config.sh script from serial terminal, a segmentation fault is received.
Resolved	IPY00029922	35353	232	Host Admin	The listboards -l2 utility does not correctly assign the physical slot of Dialogic® IPT Boards. The output shows it as unknown. For details on the physical slot, the <i>pmac.cfg</i> file contains the correct values.
Resolved	IPY00029921	36670	208	Host Admin	When running CFG, it failed to save the configuration for the Dialogic® D/4PCIUF Board.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00028606	35713	--	Host Admin	Calling SRLGetSubDevicesOnVirtualBoard() to retrieve subdevices on a virtual board will sometimes result in "error = 0x64" causing application initiation process to fail due to corrupt device names.
Resolved	IPY00028519	35685	--	Host Admin	When calling GetClockingAgents() to retrieve the current TDM configuration, the results are a core dump, causing application to abort.
Resolved	IPY00028399	35860	189	Host Admin	When editing the snmpd.conf file manually or using dlgsnmpconf utility, the board server does not properly start and stop. The board server remains disabled. Workaround: Use config.sh to configure SNMP or manually tag SNMP as configured by creating empty file /usr/dialogic/cfg/.SNMPCFG file.
Resolved	IPY00011018	36402	189	Host Admin	The dlglHidentOperStatus OID shows Dialogic® Springware Boards that downloaded in the "failed" state.
Resolved	IPY00010718	35554	171	Host Admin	When one of the PCD files for the Dialogic® DM/V2400A-PCI Board is downloaded, the following warning message may appear on the console: =====< BOARD HARDWARE VERIFICATION 1=====
Resolved	IPY00010434	35842	189	Host Admin	When performing a single board stop, the SNMP tool (dlglHidentOperStatus OID) reports board in the failed state when the board is actually in the stopped state. Workaround: Run tblist to see if the board does not show in the list. Therefore, the board is really in stopped state.
Resolved	IPY00008347	28278	--	Host Admin	When specifying ParameterFile, it causes an error when starting Dialogic® services.
Resolved	IPY00008031	29850	--	Host Admin	Sigbuffer, Player, Recorder, Tonegenerator missing after download of 4x2_isdn_* loads.
Resolved	IPY00007550	35213	--	Host Admin	When stopping and then restarting the Primary Master, a segmentation fault occurs in the clocking daemon in a mixed Dialogic® DM3 and Dialogic® Springware configuration.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00032054	36657	205	Host Drivers	The drvload script runs 'cat /sbin/lsmmod' to determine whether the driver is loaded, but this will never return success as it is looking inside the lsmmod binary.
Resolved	IPY00029999	36817	205	Host Drivers	Error messages are displayed during dlistart because the drvload script attempts to load the Dialogic® Springware sctmr and dvbm drivers even though they are already loaded.
Resolved	IPY00028208	36655	198	Host Drivers	When running the Hot Swap Kit (HSK), there are issues with the install script and patch file.
Resolved	IPY00007196	36519	198	Host Drivers	The custom version of LiS included contains 6 Dialogic header files. Since LiS is a GPL-licensed package, the inclusion of these header files has caused them to be GPL.
Resolved	IPY00011021	36408	189	Host Install	The silent uninstall prints 2 messages on the screen, meaning that it is not completely silent.
Resolved	IPY00035831	--	239	Host Library	Segmentation fault occurs in libipm_vsc.so when calling gc_close() on Global Call (IP based) line device.
Resolved	IPY00028585	36439	198	Host Library	RESTART messages change the maintenance state of channel if the channel was IN SERVICE when message arrives.
Resolved	IPY00028547	35670	208	Host Library	PDK protocol delivers DETECTED/OFFERED event to the channel even if gc_ReleaseCall() was never called to clean up the previous call on this channel.
Resolved	IPY00028352	34433	198	Host Library	gc_MakeCall() fails to return any event while using the pdk_ar_r2_io protocol on Dialogic® DM3 E1 Quad Span Boards.
Resolved	IPY00010854	36123	189	Host Library	When running Global Call call progress CEPT tests, the tests fail with the following invalid termination reason: 0x570 (expected 0x542).
Resolved	IPY00006827	36644	208	Host Library	When the gc_Open() of analog voice device is attempted and an invalid voice device name is specified, the gc_Open() fails indicating that dt_open did not work.
Resolved	IPY00006712	36790	208	Host Library	No GCEV_MEDIADETECTED event is received when the first sound heard after a connect is a SIT tone.
Resolved	IPY00030667	35861	189	Host Runtime	Streaming to board takes longer than expected on Dialogic® DM/V1200A Boards. While running 1K streaming to board tests with ml2_qs2_r2mf load on DM/V1200A Boards, the following error message was seen: qStreamGroupClose(): Unbind failed fd 34, bindhandle 238, token 0x2200ee

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00038849	--	262	Host Runtime Libraries	When opening channels asynchronously with gc_open() , sequentially one after another channels fail to open.
Resolved	IPY00010661	33512	--	Host Runtime Libraries	The libdm3dxx library has segmentation fault when the function dx_setevtmask() is called to set on a fax channel.
Resolved	IPY00010257	33668	--	Host Runtime Libraries	On shut down, the Logger object has segmentation fault.
Resolved	IPY00007551	30437	--	Host Runtime Libraries	When dstart is issued, a segmentation fault can occur when starting cheetah layer.
Resolved	IPY00007326	28097	--	Host Runtime Libraries	By setting up different IOTT blocks within "user I/O" a TDX_ERROR event occurs after using the function play.
Resolved	IPY00038477	--	260	Installation	install.sh fails to detect the version of GCC on some non-English versions of Linux operating system distributions, and the installation is aborted.
Resolved	IPY00038391	--	272	IP	Dialogic® DM/IP Board stops returning events due to a DSP failure.
Resolved	IPY00038190	--	267	IP	When running high volume load tests with Dialogic® DM/IP601 cPCI boards to test SIP call control and media activity, it appears that firmware component is experiencing intermittent data access exceptions from "Task:0x1993418 StatesTask." During this time, all active calls get suspended, and performing media activity is not transmitted across the network to other end-point.
Resolved	IPY00036965	--	247	IP	The Runtime Trace Facility (RTF) does not log entries for the GC library module when running a Global Call IP-based application on Service Update 241.
Resolved	IPY00033563	--	239	IP	SIP to SIP speech path is broken between 180 Ringing(SDP) and 200 OK.
Resolved	IPY00010929	36497	239	IP	GCEV_SERVICERESP is not received by gc_ReqService() function.
Resolved	IPY00044700	--	296	IP Host	An incorrect processing of certain H.323 parameters in a call disconnect caused a library exception condition that trickled up to the customer application as a crash.
Resolved	IPY00044544	--	296	IP Host	Placing a SIP call that sends an INVITE message with certain length SIP diversion header field contents results in a core dump. This only occurs when the gc_h3r RTF logging module is enabled.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00043307	--	294	IP Host	After performing a peripheral hot swap of Dialogic® DM/IP Boards, and boards are successfully restarted, gc_OpenEx() fails to open on-board IPML devices, returning error 0x44, Invalid Parameter.
Resolved	IPY00042528	--	284	IP Host	An error case in an IP Host library allowed the processing of certain bad messages that lead to a process crash under certain SIP application on Dialogic® IPT Boards.
Resolved	IPY00010760	36647	239	IP Host	When calling an invalid IP address or a valid one that is not answering SIP calls, you have to wait 64 seconds before you can call gc_DropCall() function.
Resolved	IPY00044273	--	294	IP Media Session Control	A request to re-INVITE from audio to T.38 fax using gc_ReqModifyCall() in IP_T38_MANUAL_MODIFY_MODE fails with GCEV_MODIFY_CALL_FAIL event.
Resolved	IPY00040439	--	272	IP Media Session Control	Client fails to join the Multicast group for certain Multicast group IP addresses when using Multicast client feature on Dialogic® IPT Boards.
Resolved	IPY00006077	36327	243	IP Media Session Control	Unexpected RTCP Goodbye messages occur during a call.
Resolved	IPY00028380	35615	171	IP Protocols	When a Global Call H.323 application fails to register with Gatekeeper, the subsequent registrations will continue failing to register properly even if the first registration was fixed.
Resolved	IPY00079365	--	309	IPT Boards	A Dialogic® IPT Board channel locked when it failed to connect to the remote media address. The problem occurred when the application made an outbound IP call from the IPT cPCI host, and the remote carrier responded with an 18x (SDP) containing their remote media IP information. If that remote media IP address is unreachable, the IPT Board does not post either the GCEV_EXTENSION events with the media connection details, nor does it post a subsequent (and required) GCEV_CONNECTED. The channel becomes unusable, and a reboot is required.
Resolved	IPY00044779	--	296	IPT Boards	gc_OpenEx() fails with error message "IPERR_INTERNAL" following a successful single board stop, removal, re-insertion, and single restart of a Dialogic® IPT Board; meanwhile, a second board was actively processing calls throughout the hot swap process on the former.
Resolved	IPY00044199	--	294	IPT Boards	After performing a peripheral hot swap of Dialogic® IPT Boards, and boards are successfully restarted, gc_OpenEx() does not return a result and causes the application to hang.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00043801	--	294	IPT Boards	The PMAC transport RTF logging module generates multiple "error 995" messages during system startup when Dialogic® IPT Boards are in the system.
Resolved	IPY00043292	--	290	IPT Boards	SIP hairpin call fails with no audio sent from Dialogic® IPT4800C IP Board. The IPT4800C does not continue the transmission of RTP and RTCP packets when ICMP is received from remote SIP client.
Resolved	IPY00043267	--	290	IPT Boards	Audible clicking is heard in the audio stream on a system using Dialogic® IPT4800C IP Boards when the IPM resource was not listening to a CT Bus time slot; the silence pattern was not set properly according to the time slot encoding, instead the default PCM silence pattern 0x00 was transmitted.
Resolved	IPY00042464	--	287	IPT Boards	Jitter alarms are repeatedly set ON and persist after the call is ended.
Resolved	IPY00042336	--	287	IPT Boards	RTP streaming from a Dialogic® IPT Board to a SIP client stops after approximately 3 seconds if the client does not send RTCP.
Resolved	IPY00042300	--	284	IPT Boards	Dialogic® IPT Board will not switch to T.38 if it rejected a previous audio re-INVITE.
Resolved	IPY00042204	--	284	IPT Boards	When using Dialogic® IPT Board, IP channel signaling APIs (gc_MakeCall() , gc_AnswerCall() , etc.) would fail, and RTF logs indicate that local media information is invalid. Subsequent calls can only be completed successfully after a gc_ResetLineDev() on that channel.
Resolved	IPY00041296	--	278	IPT Boards	Outgoing H.323 calls result in GCEV_TASKFAIL at the Alerting stage. After the call's release, no successful call can be made on that channel (iptB1Tx, ipmB1Cx device) until the board is reset.
Resolved	IPY00040743	--	275	IPT Boards	In order to achieve a performance and debugging capabilities balance with Dialogic® IPT Boards, some logging considered excessive has now been eliminated.
Resolved	IPY00040179	--	278	IPT Boards	Cannot open a full (480 channel) Dialogic® IPT Board of 1PCC IP devices.
Resolved	IPY00039847	--	271	IPT Boards	Sending of a fax is unsuccessful when making a TDM fax call across two IPT machines joined by H.323 across an IP network. RequestMode is not triggered upon CED detection in AUTO mode.
Resolved	IPY00039248	--	267	IPT Boards	GCEV_SERVICERESP event is not sent to the application when calling gc_ReqService() in ASYNC mode, even when SIP registration succeeds.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00038060	--	260	IPT Boards	An assert occurs when there are no media attributes containing "rtpmap" in a SIP INVITE to a Dialogic® IPT Board.
Resolved	IPY00037401	--	260	IPT Boards	The ipm_Ping() function fails to report packet lost when pinging an invalid IP address on the network from a Dialogic® IPT4800C Board.
Resolved	IPY00033763	--	243	IPT Boards	Dialogic® IPT Boards are time-stamping packets incorrectly.
Resolved	IPY00033164	--	243	IPT Boards	Lost Packet Alarm is not working on Dialogic® IPT Boards.
Resolved	IPY00031591	36727	205	IPT Boards	When running on a system with a Dialogic® IPT Board, the DROPCALL event is missed. After the missed DROPCALL event, it issues a gc_ReleaseCall() with the following error message: Function not supported in this state.
Resolved	IPY00031546	36833	205	IPT Boards	Unexpected IPMEV_LISTEN and IPMEV_UNLISTEN (0x9006 and 0x9007) events are reported in the system.
Resolved	IPY00028460	36298	239	IPT Boards	An extra "reserved" codec value is sent in the INVITE if coders set to "don't care."
Resolved	IPY00028222	36483	239	IPT Boards	IPT assert occurs under traffic.
Resolved	IPY00010671	35923	189	IPT Boards	T.38 on Dialogic® IPT Boards is limited to RFC2833 and G.711 pass through. An error is reported during T.38 close when used in Inband mode even though the T.38 transmission is successful.
Resolved	IPY00037319	--	257	JCT Call Control	If a board running ISDN 4ESS receives a CALL PROGRESS message in which the LOCATION information element in the Progress Indicator is 1010 - Location (network beyond interworking point), it sends back a STATUS message to the switch with Cause Value 100 (Invalid Information Element Contents).
Resolved	IPY00038551	--	271	Modular Station Interface (MSI)	ms_stopfn() causes two TSC_MsgReleaseCall messages to be sent to the DM3 Analog TSP.
Resolved	IPY00038433	--	267	Modular Station Interface (MSI)	The ms_stopfn() function fails to stop the ringing on Dialogic® DISI32R2 Board.
Resolved	IPY00028386	35532	189	Non-Facility Associated Signaling (NFAS)	The maximum number of trunks that can be controlled with one D-channel is 10. In addition, the NFAS functionality is not supported across multiple Dialogic® DM/V-B and Dialogic® DMN160TEC Boards. In other words, with these two boards, NFAS can only share the D-channel if the trunks all belong to the same board.
Resolved	IPY00042299	--	283	OA&M	For C shell (CSH) users, installation scripts fail because of syntax error in <i>ct_intel.csh</i> .

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00040536	--	284	OA&M	Some client-server based Dialogic® components (e.g., device mapper, event service) would crash when a port scan was performed on a running Dialogic® system.
Resolved	IPY00038280	--	260	OA&M	A non-OAMIPC based client was attempting connection to an internal software component, an OAMIPC-based server. This caused the internal OAMIPC-based server to crash when invoking the Dialogic® system service startup or shutdown.
Resolved	IPY00037437	--	247	OA&M	Opening and closing Dialogic® devices multiple times and repeatedly within the same program exposed a memory leak in the library stack that caused the program to hang eventually.
Resolved	IPY00074292	--	315	PBX Call Control	The application receives TDX_UNKNOWN and TDX_ERROR messages, eventually causing RNA on all ports.
Resolved	IPY00040903	--	273	PBX Call Control	File descriptors are being left open for each new call that is made and used in conjunction with the d42_gtcallid() function. After time (depending on total amount of memory), this causes the system to fail.
Resolved	IPY00040902	--	284	PBX Call Control	Failure to transition hook state when attempting to complete transfers on Mitel PBX. On some occasions, the result of these failures is that the PBX stops delivering calls to the Dialogic® D/82JCT-U Board.
Resolved	IPY00039014	--	268	PBX Call Control	Adept display parser cannot handle large displays correctly. Displays larger than 24 characters (per line) do not parse correctly (regardless of rules created). Nortel PBXs can be configured to use displays larger than 24 characters per line (e.g., 32 characters). When the customer does so, the functions d42_gtcallid() and d42_gtcallidex() return invalid displays.
Resolved	IPY00038866	--	297	PBX Call Control	The displays returned from d42_displayex() on Mitel are not formatted properly and are incomplete.
Resolved	IPY00044251	--	298	PBX Integration	A firmware assert occurred on the Dialogic® D/82JCT-U PBX Integration Board, when setting the MWI on a Nortel Norstar phone system PBX.
Resolved	IPY00037905	--	279	PBX Integration	The D4BD_M1_DISPLAYWIDTH parameter, which allows boards to support extended displays that can be provided by the M1 PBX, was not included in the <i>d42lib.h</i> header file and documentation.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00078576	--	309	PDK	<p>When the Brazil R2 Bidirectional protocol (pdk_br_r2_io) is configured for overlap send and the switch wants "silence" instead of "F" to represent end of DNIS, the ANI digits aren't sent after DNIS communication completes. When the first ANI digit was requested, no digits were sent. The protocol then timed out and the call failed.</p> <p>Note: A new parameter, CDP_SKIP_A3_AND_A4_PULSE, has been added to the <i>pdk_br_r2_io.cdp</i> file to handle this situation. For information about the new parameter, see the Documentation Updates section for the Dialogic® Global Call Country Dependent Parameters (CDP) for PDK Protocols Configuration Guide.</p>
Resolved	IPY00041808	--	288	PDK	When running with any PDK protocol and the application is abruptly shut down, the existing calls are not dropped; furthermore, noise is heard at the terminating end due to in-band non-idle data being transmitted as a result of media exit sequence.
Resolved	IPY00037923	--	262	PDK	Using PDK protocols on a system with Dialogic® Springware Boards and Dialogic® DM3 Boards, T1/E1 GC Alarm Condition: evt=0x832 occurs, causing a GCEV_BLOCKED event. The channel remains in a BLOCKED state.
Resolved	IPY00010172	34732	--	Protocols	Some ISDN cause codes cannot be set using gc_DropCall() .
Resolved	IPY00007236	30131	--	Protocols	GCEV_DROPCALL does not return after calling gc_DropCall() .
Resolved	IPY00079477	--	311	PSTN	In an NFAS configuration, both the RESTART (RX) and RESTART ACKNOWLEDGE (TX) in isdntrace have different Interface IDs.
Resolved	IPY00080244	--	315	PSTN Call Control	A memory leak occurs while receiving H.323 calls with a Global Call-based application.
Resolved	IPY00079866	--	315	PSTN Call Control	No response is sent back to the application upon receipt of a user-to-user service 1 or 2 request.
Resolved	IPY00079825	--	315	PSTN Call Control	When receiving an IAM with the continuity check indicator set to spare (illegal value), the application handles it as if a "continuity check required" indicator was received.
Resolved	IPY00079797	--	312	PSTN Call Control	Nothing happens on the line after calling the gc_MakeCall() function.
Resolved	IPY00042609	--	284	PSTN Call Control	For GCRV_PROTOCOL errors, some of the error conditions are not given in the header file (<i>pdkerror_list.h</i>).

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00042601	--	284	PSTN Call Control	Segmentation fault happens when gc_util_insert_parm_val() is called before gc_Start() . Instead of a segmentation fault, the error should be reported gracefully.
Resolved	IPY00039179	--	271	PSTN Call Control	During a glare scenario, a GCEV_RELEASECALL event is incorrectly returned to the synchronous function gc_ReleaseCall() . Events should only be returned from asynchronous functions.
Resolved	IPY00038979	--	267	PSTN Call Control	The pdk_sw_e1_fxs_io protocol does not forward the correct reason when a call is disconnected due to detection of a SIT. The reason should indicate that SIT was detected.
Resolved	IPY00038612	--	267	PSTN Call Control	When calling gc_BlindTransfer() synchronously, the function sometimes returns -1 and takes approximately 30 seconds to return with this error.
Resolved	IPY00038244	--	260	PSTN Call Control	If gc_MakeCall() is called with GC_PARM_BLK set to NULL, ERR1 is shown in the RTF log.
Resolved	IPY00037633	--	248	PSTN Call Control	gc_BlindTransfer() does not work when using the pdk_sw_e1_ssls_io protocol with Dialogic® Springware Boards.
Resolved	IPY00037607	--	251	PSTN Call Control	If another call comes in between a gc_DropCall() and gc_ReleaseCallEx() , the call is not detected. The problem occurs when the drop call and release call are issued within 1-2 seconds of each other.
Resolved	IPY00036886	--	248	PSTN Call Control	The call type information is incorrectly being encapsulated in the METAEVENT's extevtdatap pointer in the GCEV_OFFERED event when using ISDN call control on DM3.
Resolved	IPY00036504	--	245	PSTN Call Control	Calling gc_MakeCall() causes a SETUP message to be sent. If the first response from the other side is CONNECTED , the board responds with CONNECT_ACK , but GCEV_CONNECTED is not sent to the application. The problem only occurs if the board is set to Network End; if the board is set to User End, GCEV_CONNECTED is sent.
Resolved	IPY00036448	--	243	PSTN Call Control	With 5ESS ISDN on Dialogic® Springware Boards, call setup fails when the CALLED NUMBER TYPE is set to NETWORK_SPECIFIC (0x03) .
Resolved	IPY00036347	--	245	PSTN Call Control	QERROR_WARNING messages appear in <i>Dm3StdErr log</i> . Eventually, gc_SetChanState() fails on all channels, and all channels are blocked.
Resolved	IPY00036337	--	243	PSTN Call Control	With 5ESS ISDN on Dialogic® DM3 Boards, call setup fails when the CALLED NUMBER TYPE is set to NETWORK_SPECIFIC (0x03) .

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00036248	--	243	PSTN Call Control	When using Global Call SS7, the 0xb and 0xc address signals, which were previously reported to the app as "b" and "c", are now getting reported as "#" and "**", thus breaking backward compatibility.
Resolved	IPY00036247	--	241	PSTN Call Control	A Dialogic® JCT Board running with the NT1 protocol receives an Alerting message with incorrect GCEV_PROCEEDING event instead of the expected GCEV_ALERTING on a channel.
Resolved	IPY00036044	--	241	PSTN Call Control	Failures seen when invoking gc_SetChanState() on Dialogic® JCT Boards.
Resolved	IPY00035451	--	243	PSTN Call Control	The gc_OpenEx() function fails for device ":N_dkB1T1" for Dialogic® SS7 Board when configured for clear channel.
Resolved	IPY00035148	--	243	PSTN Call Control	The gc_UnListen() function has no effect when issued on "dk" devices using Global Call SS7.
Resolved	IPY00034816	--	243	PSTN Call Control	SIT tone is not detected when using the Nortel Meridian protocol on Dialogic® DM3 Boards.
Resolved	IPY00034738	--	232	PSTN Call Control	Call progress analysis does not properly report fax tone when parameter All INTEGER_t CDP_OUT_ConnectType has a value of "1".
Resolved	IPY00034618	--	248	PSTN Call Control	gc_DropCall() fails when responding to a GCEV_DISCONNECT event after a GCEV_BLOCKED event.
Resolved	IPY00034606	--	232	PSTN Call Control	While issuing a make call during a supervised transfer to a destination that is busy, gc_ResultMsg() returns with PROTOCOL ERROR.
Resolved	IPY00034404	--	243	PSTN Call Control	In Global Call SS7, initial alarm conditions are not propagated up to application.
Resolved	IPY00033698	--	232	PSTN Call Control	The primary call can not be re-transferred via gc_SetupTransfer() when the transferred call is disconnected after SwapHold.
Resolved	IPY00032803	--	241	PSTN Call Control	Unable to make outbound calls on Dialogic® JCT Boards with DMS protocol; the 'Interface ID Present' was being changed from 0 to 1 and the 'Interface Identifier' byte was added.
Resolved	IPY00032794	--	241	PSTN Call Control	Dialogic® JCT Board rejects incoming calls in the first unblocked channel when a call is disconnected, but not released. This symptom was seen on incoming SETUP message with Channel ID IE specifying "ANY CHANNEL" and "host controlled release" feature is ON.
Resolved	IPY00006790	35137	243	PSTN Call Control	For outbound Global Call SS7 calls with dial string *1234, the leading * is stripped and replaced with a trailing 0 (i.e., 12340), causing the call to fail.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved			171	Redundant Host	Redundant Host (RH) functionality is not supported in this release.
Resolved	IPY00039492	--	268	Runtime Trace Facility (RTF)	RTF logging has a memory leak and drops some log messages.
Resolved	IPY00038545	--	268	Runtime Trace Facility (RTF)	In RTFManager, the <i>RtfMatrix.xml</i> file was used to map the modules in the RTFConfig file to a family and technology group. But if any changes were made to the RTFConfig file outside of RTFManager, the configuration section would fail. Note: The mapping file was removed, and attribute tags were added to the RTFConfig file to define the mappings, making the configuration section of RTFManager more robust.
Resolved	IPY00038524	--	260	Runtime Trace Facility (RTF)	Multiple threads can be created in the RTF server for a single client when the system is heavily loaded. This leads to a build-up of threads in the server, which can lead to thread creation failures.
Resolved	IPY00036919	--	248	Runtime Trace Facility (RTF)	Unable to configure RTF trace capabilities using RTFManager because the selection is grayed out.
Resolved	IPY00036856	--	248	Runtime Trace Facility (RTF)	Some modules in RtfConfigLinux.xml have incorrect module names. The modules are names of Windows® libraries, not Linux libraries.
Resolved	IPY00032248	36550	204	SELinux	Users running Red Hat AS 4.0 Update 2 with SELinux enabled and a targeted policy in use will encounter problems when the Dialogic® start script runs at boot time. The user will be prompted multiple times during the script, making unattended system boot impossible. This is due to the new default security policy present in AS 4.0 Update 2. Workaround: Edit the <code>/usr/dialogic/bin/intel_functions</code> script, making the following change to line 229: ORIGINAL: <code>su -s /bin/bash - \${user} -c "\${PROG} &>/dev/null &"</code> MODIFIED: <code>/sbin/runuser -s /bin/bash - \${user} -c "\${PROG} &>/dev/null &"</code>

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00032052	36656	205	SELinux	<p>Systems that have the DLGCdev RPM installed will experience a kernel panic once an attempt is made to load the Dialogic® Springware drivers. If you have not selected "Intel Dialogic Boards" from the installation menu (DLGCdev RPM not present on the system - verify by running 'rpm -q DLGCdev'), then it will allow unattended automatic startup at boot time as expected. This problem is due to interactions between the Linux Streams (LiS) software and the SELinux security policy.</p> <p>Workaround: Change the security mode from enforcing (which causes the panic) to permissive. The impact is that security issues will still be logged, but the policy will not actually be enforced.</p> <p>The steps to make this change are as follows:</p> <ol style="list-style-type: none"> 1. If Dialogic® System Release 6.1 for Linux is already installed and kernel panics are occurring during system boot, reboot the system into single-user mode. If the System Release is not yet installed, skip this step. 2. Execute "/usr/sbin/setenforce 0." This will change the security mode of the currently running kernel to permissive, but will be lost upon reboot. 3. Edit the /etc/selinux/config file, changing "SELINUX=enforcing" to "SELINUX=permissive." This will change the security mode that is set at boot time. 4. If the system was booted into single-user mode, type 'exit' to continue with the normal boot process. <p>To determine the SELinux security mode, execute "/usr/sbin/sestatus grep mode."</p>
Resolved	IPY00079251	--	309	SIP Call Control	<p>The SIP Allow header does not include the "INFO" method. When acting as a SIP User Agent Server (UAS), 1xx and 200_OK messages do not include INFO as a listed method in the Allow header. When acting as a SIP User Agent Client (UAC), INVITE does not include the INFO method in the Allow header.</p> <p>Note: A documentation update has been added in the Documentation Updates section for the Dialogic® Global Call IP Technology Guide. The INFO method is now included as part of the Allow header in SIP messages by default.</p>
Resolved	IPY00078519	--	304	SIP Call Control	<p>The application sets a 5.3 Kbps bit rate for the G.723.1 transmit codec; however, the 6.3 Kbps rate for the codec is RTP transmitted instead, and reported as such by IPPARM_FASTSTART_CODER event.</p>

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00041300	--	276	SIP Call Control	SIP calls are rejected with a "486 Busy Here" due to incorrect handling of the scenario when a previous call was terminated due to bad incoming SDP.
Resolved	IPY00041118	--	276	SIP Call Control	The application is unable to make a SIP call using the gc_MakeCall() function on the same channels previously used to make an H.323 call.
Resolved	IPY00039965	--	271	SIP Call Control	Outbound IP call fails with "IPEC_SIPReasonStatus503ServiceUnavailable" when the hostname is passed as the destination address in the dialstring. The outbound call using gc_MakeCall() is not able to resolve the hostname to an IP address for the call to complete successfully.
Resolved	IPY00039707	--	271	SIP Call Control	Automatic SIP re-INVITE when media switches from audio to fax causes a glare condition that disconnects the call. Note: To resolve this issue, new Global Call parameters have been added to disable/enable the sending of the automatic SIP re-INVITE message upon media switch. For information about this feature, see Section 1.13, "Disabling Automatic re-INVITE Message when Switching between Fax and Audio" , on page 51.
Resolved	IPY00039401	--	271	SIP Call Control	The "Record-Route" field of a SIP header message is incorrectly reported as the "Route" field when present within an incoming SIP message through use of the Dialogic® Global Call API.
Resolved	IPY00038572	--	262	SIP Call Control	When running a Dialogic® Global Call IP-based application that enables notification of SIP messages through GCEV_EXTENSION events, the type of SIP message received with the event cannot be identified. The message type value retrievable with that event returns more bytes than expected, making it unable to decipher which message was received.
Resolved	IPY00038365	--	260	SIP Call Control	Egress SIP calls work briefly, but then omit SDP in egress INVITE message.
Resolved	IPY00038240	--	260	SIP Call Control	An assert occurs on inbound re-INVITE.
Resolved	IPY00035822	--	239	SIP Call Control	Global Call SIP application does not respond to 407 Proxy Authentication Required messages.
Resolved	IPY00034627	--	239	SIP Call Control	Format Specific Parameters of the Media Format Attribute are not sent within SDP body of an Invite message when using non-default value for IPPARM_DTMF_RFC2833_PAYLOAD_TYPE.
Resolved	IPY00034406	--	239	SIP Call Control	The handle count of the application is high as compared to the previous Service Updates. The handle count used by application is around 20000.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00033912	--	239	SIP Call Control	Early media on 180 Ringing(NO SDP) fails if T.38 required.
Resolved	IPY00038230	--	260	SNMP	When gathering SNMP traps on a system with DS1 lines, the dsx1CurrentEntry table is populated with the incorrect index numbers.
Resolved	IPY00008148	27990	--	SNMP	If SNMP is installed, ISDN calls does not receive disconnect messages.
Resolved	IPY00043515	--	290	Springware Call Control	Outbound calls made by a Dialogic® D/480JCT-2T1 Media Board with NI2 firmware fail intermittently. The call SETUP is successful, but after receiving a STATUS message, the board sends a CALL DISCONNECT message to disconnect the call with an indication of Protocol Error.
Resolved	IPY00042862	--	287	Springware Call Control	GCEV_UNBLOCKED event doesn't arrive for individual channels, even though GCEV_BLOCKED was delivered to individual channels, after AIS alarms occur and are then cleared.
Resolved	IPY00042730	--	287	Springware Call Control	Glare condition causes calls to be dropped immediately after answering when using ISDN 5ESS protocol, firmware file <i>spis5ess.fwl</i> .
Resolved	IPY00042408	--	284	Springware Call Control	After calling gc_BlindTransfer() , application receives GCEV_TASKFAIL event followed by GCEV_BLINDTRANSFER event. The expected result is one event or the other, but not both.
Resolved	IPY00042208	--	280	Springware Call Control	GCEV_PROGRESSING message with 0x02 Progress Description is not detected when using the DMS protocol.
Resolved	IPY00042003	--	283	Springware Call Control	The METAEVENT associated with GCEV_DETECTED always returns 0 as a CRN value; it does not return the correct CRN.
Resolved	IPY00041987	--	283	Springware Call Control	gc_BlindTransfer() fails to return any events in failing blind transfer scenarios.
Resolved	IPY00041959	--	283	Springware Call Control	Intermittent problem when trying to retrieve user attributes with gc_GetMetaEvent() .
Resolved	IPY00041792	--	279	Springware Call Control	Incorrect cause code is sent to application upon GCEV_DISCONNECT event. The cause code is 'non-selected user clearing' while link trace shows 'normal clearing' instead.
Resolved	IPY00039331	--	268	Springware Call Control	When using DPNSS, the response to the setup message from the switch is incorrect; an incomplete Number Acknowledge Msg is returned.
Resolved	IPY00039249	--	271	Springware Call Control	When gc_WaitCall() is issued after an incoming call is pending, the gc_AcceptCall() fails even though the application receives the GCEV_OFFERED event.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00038539	--	262	Springware Call Control	Interface ID Present enabled in network setup message causes channel on Dialogic® D/480JCT Board using 4ESS protocol to reject call with invalid information element contents.
Resolved	IPY00038130	--	262	Springware Call Control	A GCEV_FATALERROR occurs on Dialogic® D/480JCT-2T1 Board.
Resolved	IPY00078799	--	307	Springware CSP	Automatic gain control (AGC) was purposely disabled when using CSP to avoid audio samples alteration. As a consequence, AGC had no effect on the audio level streamed with the ec_stream() or equivalent function. AGC is now enabled with Dialogic® Springware Boards only when AGC is explicitly enabled.
Resolved	IPY00043965	--	293	Springware Drivers	A kernel panic occurred due to corrupt message going to the device driver.
Resolved	IPY00042650	--	283	Springware Drivers	After a period of time, the IRQ that is assigned to the Dialogic® D/82JCT-U Board gets disabled, and the board no longer processes calls. The problem occurs on a HP Server running SUSE Linux Enterprise Server 10 and using 8259 PIC mode.
Resolved	IPY00045074	--	304	Springware Fax	Improper retraining during error correction mode (ECM) receive with noisy phone lines would cause the Dialogic® VFX/41JCT-LS Media Board to keep requesting the same set of ECM frames to be transmitted; thus, the time to receive a faxed document became very long.
Resolved	IPY00042940	--	287	Springware Fax	Channel on Dialogic® VFX/41JCT-LS Board stays in "fax receiving" status due to improper handling of a "busy tone" although the call is disconnected. The hung channel does not go back to idle state unless application calls fx_stopch() .
Resolved	IPY00042752	--	287	Springware Fax	GTD was disabled during a fax receive session (fx_rcvfax()), thus preventing the disconnect tone resulting from a hang-up on the far end being detected by the voice channel, leaving the line busy until the fax receive idles out. This can occur, for example, when a call is made by a voice caller who hangs up after hearing the fax receive tones.
Resolved	IPY00040781	--	272	Springware Fax	Application crashes when it is about to finish receiving a fax when using Dialogic® D/82JCT-U Boards.
Resolved	IPY00039341	--	268	Springware Fax	The Dialogic® VFX/41JCT-LS Media Board sometimes fails to receive fax with ATFX_ESTAT() = 195 when using 14.4kbps/no ECM mode and multi-page signal (MPS).
Resolved	IPY00038836	--	262	Springware Fax	Fax error codes are not reported properly with Dialogic® VFX41JCT-LS Board.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00038298	--	260	Springware Fax	When using Dialogic® VFX/41JCT-LS Board, multiple consecutive ECM fax receive calls failed and ATFX_ESTAT() reported 198. Non-ECM fax receives by the same channel were successful.
Resolved	IPY00036054	--	248	Springware Fax	The EFX_UNSUPPORTED error is returned when trying to send a TIFF file created using a third party utility with the Dialogic® VFX/41JCT Board.
Resolved	IPY00034495	--	241	Springware Fax	Firmware crash occurs when certain TIFF file is sent from one channel in MH, 9600 MSLT=10ms condition.
Resolved	IPY00045136	--	298	Springware Firmware	When using Dialogic® D/120JCT Media Board under heavy CSP channel utilization, a firmware assert occurred.
Resolved	IPY00043029	--	289	Springware Firmware	When using CTR firmware with Dialogic® D/600JCT-2E1 Media Board, one or two ports become inoperable and are unable to answer incoming calls under heavy load.
Resolved	IPY00037483	--	251	Springware Firmware	Firmware assert during load test causes boards to stop responding to driver.
Resolved	IPY00037318	--	251	Springware Firmware	Dialogic® Springware Board ISDN 4ESS protocol does not support LOCATION type 1010 in Progress Indicator.
Resolved	IPY00036833	--	251	Springware Firmware	When using NI2 protocol on Dialogic® JCT Boards, disconnect glare causes next call to be rejected with cause code 44, channel not available.
Resolved	IPY00036830	--	251	Springware Firmware	The DPNSS cause "Network Termination" (NT=0x02) is not supported.
Resolved	IPY00033185	--	241	Springware Firmware	On Dialogic® Springware Board ISDN 5ESS and 4ESS protocols, loopback calls from user to network fail.
Resolved	IPY00045478	--	304	Springware ISDN	When using NI2 protocol on Dialogic® Springware Boards, and a RELEASE COMPLETE is not returned by the switch after a successful call, improper handling of T308 timer expiration would cause any subsequent incoming calls to fail on that channel, hanging on gc_AnswerCall() .
Resolved	IPY00035506	--	241	Springware ISDN	An ISDN call disconnects during the ACCEPT state. When this occurs, the application does not get a CCEV_DISCONNECT event.
Resolved	IPY00041345	--	278	Springware ISDN Firmware	Firmware assert occurs due to zero length User-User IE message, and Dialogic® board stops responding to the switch.
Resolved	IPY00036665	--	257	Springware Network	Missing DROPCALL event - disconnect incomplete - spurious interrupt firmware crash in Dialogic® JCT Board DPNSS firmware.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00042579	--	287	Springware PBX	Using d42_gtcalled() causes a memory leak.
Resolved	IPY00039490	--	272	Springware PBX	The d42_setparm() for the parameter 0x1A does not work on the Dialogic® D/42JCT-U Board.
Resolved	IPY00078411	--	304	Springware PSTN	When using PDK_US_LS_FXS_IO protocol on Dialogic® Springware Boards, gc_DropCall() did not generate a GCEV_DROP_CALL event after a glare condition.
Resolved	IPY00042985	--	287	Springware Voice	After upgrading to a new Service Update, DTMF digits sent from some specific phone terminal could not be detected by Dialogic® Springware Boards.
Resolved	IPY00040096	--	272	Springware Voice	Failure to increase media play speed by more than 25% when using dx_adjsv() to set the play speed; the documentation specifies a maximum change of 50%.
Resolved	IPY00040052	--	271	Springware Voice	Perfect Call call progress analysis on Dialogic® Springware Boards sometimes falsely detects dial tone and proceeds with dialing while there is no signal matching for the dial tone criteria.
Resolved	IPY00038516	--	262	Springware Voice	Using dx_setparm() to set the value of DXCH_XFERBUFSIZE always returns "Invalid Parameter."
Resolved	IPY00037542	--	248	Springware Voice	Calls to Dialogic® functions change the syslog program from whatever the customer has set (var/log/messages). It appears that each Dialogic® function call from the Voice API calls openlog() even though messages from the Dialogic® libraries are not seen.
Resolved	IPY00008747	31924	--	Springware Voice	A segmentation fault results when user-defined I/O (DX_UIO) is used, and a WAVE is specified as the XPB file format when calling dx_reciottdata() .
Resolved	IPY00008202	27163	--	Springware Voice	When an application issues ec_reciottdata() or ec_stream() on a CSP capable device and then plays a prompt using dx_playiottdata() , there's a glitch that can be heard at the beginning of the play.
Resolved	IPY00008014	31925	--	Springware Voice	If DX_UIO struct is defined, any call to dx_reciottdata() uses the UIO even if not specified in the IOTT struct.
Resolved	IPY00045239	--	300	SS7	The small window of time between the receipt of a GCEV_UNBLOCKED and gc_WaitCall() completion was enough to miss GC/SS7 calls; the library discarded calls received during that window.
Resolved	IPY00045224	--	300	SS7	Dialogic® Global Call SS7 application does not work properly the first time after the Dialogic® SS7 Boards are initialized; the application needs to be torn down and brought up again for it to work properly.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00044425	--	298	SS7	A GCEV_OFFERED event was sent before the application issued gc_WaitCall() ; this is not the correct call flow order in Global Call.
Resolved	IPY00044100	--	298	SS7	The GC SS7 server log level configuration in <i>gcss7.cfg</i> is not working; whether the debug level is set to "All," "None," or "Errors," the log file is the same as "All."
Resolved	IPY00037918	--	257	SS7	The RSI link goes down intermittently.
Resolved	IPY00037767	--	251	SS7	The GCSS7 library does not generate the GCEV_MOREINFO event if it receives a SAM message with only STOP digit (0xf) after the application has already issued gc_CallAck() .
Resolved	IPY00037632	--	251	SS7	If there is a delay in the SS7 server picking up messages from the IPC queue, an ERROR_IO_PENDING occurs and the SS7 library terminates the IPC. This causes all the circuits to get blocked, as there is no more connection with the SS7 service. This is causing the IVRs to get a sudden circuit block from the switch in all of its SS7 circuits.
Resolved	IPY00008668	50087	--	SS7 Boards	If bearer trunks are disconnected when calls are being set up, the GCEV_BLOCKED event is not generated.
Resolved	IPY00043230	--	289	Standard Runtime Library (SRL)	An access violation occurs during sr_waitvtEx() processing.
Resolved	IPY00039334	--	278	Standard Runtime Library (SRL)	An application crash occurred; the stack trace shows SRL library at the top of the stack.
Resolved	IPY00039155	--	271	Standard Runtime Library (SRL)	An application crash occurs with SRL at the top of the stack; the SRL was not initializing all variables of a structure for a given thread, which can cause an access violation.
Resolved	IPY00038708	--	262	Standard Runtime Library (SRL)	An access violation occurs when application calls sr_waitvtEx() for the same device on multiple threads.
Resolved	IPY00028419	35888	171	Trunk Configurator	When creating a mixed protocol load via the trunk configurator for the Dialogic® DMT160TEC Board, a current restriction exists for the E1 clear channel protocol, E1CC. This only affects the mixed loads that contain the E1 clear channel protocol. The rest of the supported protocol combinations are still functional.
Resolved	IPY00010337	34184	--	UUI IE	UUI information is not retrieved correctly with more than 71 bytes of information in the IE_BLK.
Resolved	IPY00007895	29545	--	UUI IE	UUI IE information is not retrieved correctly with more than 71 bytes of information in the IE_BLK.

Issues Sorted By Type, Dialogic® System Release 6.1 for Linux (Continued)

Issue Type	Defect No.	PTR No.	SU No.	Dialogic® Product(s) or Component(s)	Description
Resolved	IPY00080252	--	315	Voice	Voice Media intensive (plays/records) caused the play and record functions to fail. TDX_ERROR events with reason 0x80000 (system error) were observed.
Resolved	IPY00080145	--	315	Voice	The voice channel remains in a PLAYING state after the dx_playiottdata() function returns a failure (-1) when called asynchronous mode.
Resolved	IPY00079561	--	315	Voice	ATDX_CRTNID returns a 0 instead of the proper value.
Resolved	IPY00079523	--	311	Voice	While retrieving board status, d42_getbrdstatus crashes in debug mode and returns an incorrect value in release mode.
Resolved	IPY00038435	--	260	Voice	Channels hang and are not able to recover once in a CS_STOPD state.
Resolved	IPY00041369	--	278	Voice API	The ai_open() function fails on Dialogic® DISI Boards, making it impossible to use an audio input device.
Resolved	IPY00037818	--	257	Voice API	<p>The dx_setevtmask() function fails to disable the TDX_CST events for DE_DIGITS when setting the DM_DIGOFF flag.</p> <p>Note: In Dialogic® System Release 6.1 for Linux, user defined tone digit reporting is enabled by default. The dx_setevtmask() function can be used to enable and disable call status transition (CST) events for user defined tones. A documentation update has been added in the Documentation Updates section for the Dialogic® Voice API Library Reference.</p>
Resolved	IPY00037796	--	257	Voice API	<p>TDX_RESETERR and TFX_FAXRCV events have the same value defined in their respective header files (<i>dxxlib.h</i> and <i>faxlib.h</i>). This can lead to a conflict at the application level when performing either the voice or fax channel recovery feature; you cannot tell the difference between a pair of potential completion events when executing dx_resetch() and fx_rcvfax() API calls.</p> <p>Note: The fix for defect IPY00037796 changed the values of the two defines, TDX_RESET and TDX_RESETERR, in <i>dxxlib.h</i> from 0xA1/A2 to the following:</p> <pre>#define TDX_RESET (DXDEV_BASE 0x01) // reset event for Reset API's #define TDX_RESETERR (DXDEV_BASE 0x02) // reset error event for Reset API's</pre> <p>Applications with Dialogic® DM3 Boards that use the dx_resetch() function must be recompiled to work properly.</p>

Conference Density Limitations on Dialogic® DM/IP Dual Span Boards

		Open IP Devices																			
		42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	
Connected Voice Devices	42																			30	
	43																		30	29	
	44																	30	29	28	
	45																30	29	28	27	
	46															30	29	28	27	26	
	47														30	29	28	27	26	25	
	48														30	29	28	27	26	25	
	49													30	29	28	27	26	25	24	
	50												30	29	28	27	26	25	24	23	
	51											30	29	28	27	26	25	24	23	22	
	52										30	29	28	27	26	25	24	23	22	21	
53									30	29	28	27	26	25	24	23	22	21	20		
54								30	29	28	27	26	25	24	23	22	21	20	19		
55							30	29	28	27	26	25	24	23	22	21	20	19	18		
56						30	29	28	27	26	25	24	23	22	21	20	19	18	17		
57				30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15		
58			30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14		
59		30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13		
60	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12		

Above table defines the conference density limitations on Dialogic® DM/IP Dual Span Boards (Dialogic® DM/IP601-2E1-PCI-100BT and Dialogic® DM/IP601-2E1-100cPCI) depending on the number of IP devices that are open and number of voice devices that are connected (i.e., **dx_Listen()** has been called) at a given time. Limits do not apply to DM/IP Single Span or Resource Boards, which always support the full 30 conference resources.

Note: Shaded areas support full conference density of 30.

Documentation Updates

This chapter contains information on updates and corrections to the documents included in Dialogic® System Release 6.1 for Linux. Documentation updates are divided into the following categories:

- [System Release Documentation Updates 158](#)
- [Installation and Configuration Documentation Updates 162](#)
- [OA&M Documentation Updates 178](#)
- [Programming Libraries Documentation Updates 183](#)
- [Demonstration Software Documentation Updates 197](#)
- [Pigeon Point Systems Linux Hot Swap Kit Documentation Updates 198](#)

3.1 System Release Documentation Updates

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

- [Dialogic® System Release 6.1 for Linux Release Guide](#)

3.1.1 Dialogic® System Release 6.1 for Linux Release Guide

Updates to Chapter 2, System Requirements

Update to **Section 2.2, Basic Software Requirements** (IPY00041134)

The following information currently appears in this section under **Requirements for NetSNMP**:

- SNMP may not be supported if the Linux Distribution does not use NetSNMP Master Agent 5.09. Source code for NetSNMP is included with System Release 6.1 for Linux. Do not use the NetSNMP that is included with Red Hat Linux. Refer to the *Software Installation Guide* for instructions on installing NetSNMP.

Note that Red Hat Enterprise Linux 5 and SUSE Linux Enterprise Server 10 use a NetSNMP version that is not supported with System Release 6.1 Linux, and the Dialogic SNMP modules fail to load when used with these OSDs. As a workaround, do either of the following:

- Use the version of the NetSNMP source included in the System Release 6.1 Linux distribution to build 5.0.9 binaries, which have been tested; or
- Obtain a binary RPM containing a NetSNMP version between 5.0.9 and 5.2, which can be used with System Release 6.1 Linux.

Update to **Section 2.2, Basic Software Requirements**

With the Service Update, Note 3 should be revised to say that this release now supports 12 GB RAM (no longer limitation of 4 GB RAM).

Update to **Section 2.2, Basic Software Requirements**

Since the original release of Dialogic® System Release 6.1 for Linux, additional operating system distributions (OSDs) are now supported with the Service Update. See [Section 1.37, “Additional Supported Operating System Distributions \(OSDs\)”](#), on page 90 of this Release Update. Related information about software requirements is also given in [Section 1.20, “Support for LiS 2.19.1”](#), on page 62 and [Section 1.57, “Supported Kernel and GCC Versions”](#), on page 104 of this Release Update.

Note: Only 32-bit versions of Linux OSDs are supported.

Updates to Chapter 3, New Features by Product

Update to **Section 3.2, New DMV160LP Products** (PTR# 36105)

In the list of features for Dialogic® DMV160LP Boards, the following feature (under “Global Call support”) is not supported and should be deleted:

- Hook-flash through the Global Call API

Updates to **Section 3.4, New DMT160TEC Digital Telephony Interface Products**

The following feature should be added under the “Features” heading:

- Software selectable T1/E1 (in groups of 4 spans), providing the ability to mix T1 and E1 on a board. Refer to the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* for more information.

The “call progress analysis” reference should be included in the following feature description (highlighted in bold):

- Tone detection and generation, including GTD/GTG and **call progress analysis**. Refer to the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* for more information.

Update to **Section 3.5, New DM/V-B Products**

The use of the term “selected” is incorrect in this section. The boards only have the ability to mix “select” protocols on a board. “Selected” implies any protocol selected can be mixed; however, only a specific or a “select” number of protocols can be mixed on a board. Any reference to “selected” protocols should be changed to “select” protocols.

Update to **Section 3.6, New IPT10000C Board**

The term “two and 1-Gigabit” should be revised to “two 100/1000 Base-T” in the “IPT10000C” description.

Updates to **Section 3.7, New SS7 Boards and SIUs** (IPY00081381)

Under the Features subheading, added the following note:

Note: Global Call SS7 binaries are linked with the shared library of the Dialogic® SS7 DSI Development Package. Global Call SS7 customers must use the Dialogic® SS7 DSI Development Package version 5.0 or later. If an older version is used, the Global Call SS7 server will not start during download.

Updates to **Section 3.9, New Features for JCT Products**

The note under the “Continuous speech processing (CSP) support” feature is incorrect and should be ignored. The note should state the following:

Note: CSP is supported on all Dialogic® JCT digital boards except for the D/300JCT-E1 Board.

The note under the “DPNSS support” feature is incorrect and should be ignored. The note should state the following:

Note: This feature only applies to the Dialogic® D/300JCT-E1, D/600JCT-1E1, and D/600JCT-2E1 Boards.

The note under the “NFAS (non-facility associated signaling) support” feature is incorrect and should be ignored. The note should state the following:

Note: This feature only applies to the Dialogic® D/240JCT-T1, D/480JCT-1T1, and D/480JCT-2T1 Boards. NFAS is only supported on T1 JCT Boards and does not support E1, resource, or digital boards.

Update to **Section 3.10, New Features for DM/F Products**

This section incorrectly states that the DM/F Boards support voice. These boards do not provide voice support.

Updates to **Section 3.12, New Features for DM/V and DM/V-A Products**

The following feature should be added under the “New DM/V Product Features” heading:

- Additional GCAMS (Global Call Alarm Management System) alarms, which are provided through Global Call. See the *Dialogic® Global Call ISDN Technology Guide*, *Dialogic® Global Call E1/T1 CAS/R2 Technology Guide*, and *Dialogic® DM3 Architecture Products on Linux Configuration Guide* for more information.

GCAMS is documented as a DM/V-A feature, but it is actually supported on DM/V Boards and should be listed under that section.

The use of the term “selected media loads” is incorrect in the “Ability to stream echo-cancelled data over the CT bus to another board” feature. “Selected” implies that any media load selected can be mixed; however, only specific or “select” number of media loads can be mixed. It should read “select media loads” in the description.

The “DM/V2400A-PCI (media load 9C)” and “DM/V2400A-cPCI (media load 5)” references should be included in the following feature description (highlighted in bold):

- Enhanced voice + fax media loads added for Dialogic® DM/V960A-4T1-PCI (media load 5), Dialogic® DM/V480A-2T1-PCI (media load 5BC), Dialogic® DM/V600A-2E1-PCI (media load 5BC), Dialogic® DM/V2400A-PCI (media load 5), **Dialogic® DM/V2400A-PCI (media load 9C)**, and **Dialogic® DM/V2400A-cPCI (media load 5)**. Refer to the *Dialogic® DM3 Architecture Products on Linux Configuration Guide* for more information.

Updates to Chapter 5, OA&M Software

Update to **Section 5.1, Administration Software** (IPY00041134)

The following note currently appears in this section under **SNMP Agent Administration Software Features**:

- SNMP may not be supported if the Linux Distribution does not use NetSNMP Master Agent 5.09. Source code for NetSNMP is included with System Release 6.1 for Linux. Do

not use the NetSNMP that is included with Red Hat Linux. Refer to the *Software Installation Guide* for instructions on installing NetSNMP.

Note that Red Hat Enterprise Linux 5 and SUSE Linux Enterprise Server 10 use a NetSNMP version that is not supported with System Release 6.1 Linux, and the Dialogic SNMP modules fail to load when used with these OSDs. As a workaround, do either of the following:

- Use the version of the NetSNMP source included in the System Release 6.1 Linux distribution to build 5.0.9 binaries, which have been tested; or
- Obtain a binary RPM containing a NetSNMP version between 5.0.9 and 5.2, which can be used with System Release 6.1 Linux.

Update to **Section 5.1, Administrative Software**

With the Service Update, Peripheral Hot Swap (PHS) is now supported on the following chassis/Single Board Computers (SBC) and should be documented under the subsection named “Peripheral Host Swap (PHS)”:

- Intel NetStructure ZT5085 or MPCHC5085 with MPCBL5524A1C (Dual CPU with up to 4 GB RAM)
- Intel NetStructure ZT5085 or MPCHC5085 with MPCBL5524A1D (Single CPU with up to 4 GB RAM)

Update to **Section 5.1, Administrative Software**

With the Service Update, Redundant Host (RH) is now supported on the following chassis/Single Board Computers (SBC) and should be documented under a new subsection named “Redundant Host (RH)”:

- Intel NetStructure ZT5085 or MPCHC5085 with MPCBL5524A1C (Dual CPU with up to 4 GB RAM)
- Intel NetStructure ZT5085 or MPCHC5085 with MPCBL5524A1D (Single CPU with up to 4 GB RAM)
- Intel NetStructure ZT5085 or MPCHC5085 with ZT5524A-1A (Dual CPU)
- Intel NetStructure ZT5085 or MPCHC5085 with ZT5524A-1B (Single CPU)

Updates to Chapter 8, Supported Hardware

Update to **Section 8.3, Signaling Products**

With the Service Update, the following Dialogic® DM/IP Boards are supported and should be documented in the list of supported boards for Signaling Products:

- Dialogic® DM/IP241-1T1-PCI-100BT IP Boards
- Dialogic® DM/IP301-1E1-PCI-100BT IP Boards
- Dialogic® DM/IP481-2T1-PCI-100BT IP Boards
- Dialogic® DM/IP601-2E1-PCI-100BT IP Boards
- Dialogic® DM/IP481-2T1-cPCI-100BT IP Boards
- Dialogic® DM/IP601-2E1-cPCI-100BT IP Boards
- Dialogic® DM/IP601-cPCI-100BT IP Boards

3.2 Installation and Configuration Documentation Updates

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

- [Dialogic® System Release 6.1 for Linux Software Installation Guide](#)
- [Dialogic® DM3 Architecture Products on Linux Configuration Guide](#)
- [Dialogic® IPT Series Products on Linux Configuration Guide](#)
- [Dialogic® Springware Architecture Products on Linux Configuration Guide](#)
- [Dialogic® Global Call Country Dependent Parameters \(CDP\) for PDK Protocols Configuration Guide](#)

3.2.1 Dialogic® System Release 6.1 for Linux Software Installation Guide

Updated Software Requirements

Since the original release of Dialogic® System Release 6.1 for Linux, additional operating system distributions (OSDs) are now supported with the Service Update. See [Section 1.37, “Additional Supported Operating System Distributions \(OSDs\)”](#), on page 90 of this Release Update. Related information about software requirements is also given in [Section 1.20, “Support for LiS 2.19.1”](#), on page 62 and [Section 1.57, “Supported Kernel and GCC Versions”](#), on page 104 of this Release Update.

Update to **Section 2.1.5, Preparing to Run the System Release with Variations on the Default Kernels**

Step 1 of the procedure should be changed to indicate that it applies to **all** SUSE Linux Enterprise Server OSDs (not just to SLES 9).

Update to **Chapter 4, Troubleshooting**

The following new subsection should be added to **Chapter 4, Troubleshooting**.

Upgrading the Hot Swap Kit/Redundant Host Software after Upgrading the System Release

Whenever you upgrade a Dialogic® System Release, the Hot Swap Kit and Redundant Host software should be upgraded *before* you upgrade the System Release. You could experience problems using the software if you use the latest Service Update with an old version of the Hot Swap Kit/Redundant Host software. This section describes what to do if you didn't upgrade old Hot Swap Kit and Redundant Host software before upgrading the System Release.

1. Stop all services.
2. Uninstall the Hot Swap Kit. Use `hsk_uninstall` as described in the *Pigeon Point Systems Linux Hot Swap Kit User Guide*, which is a PDF file located in the *redistributable-runtime/PPS* directory.

3. Reboot.

Note: If `ct_intel` is in the boot sequence, then there will be harmless errors during the boot session (since the driver was removed during the Hot Swap Kit/Redundant Host uninstall).

4. Install the new Hot Swap Kit. Use `hsk_install` as described in the *Pigeon Point Systems Linux Hot Swap Kit User Guide*, which is a PDF file located in the `redistributable-runtime/PPS` directory. Make sure you choose the correct Single Board Computer (SBC) and chassis.

5. Reboot into the new Hot Swap Kit/Redundant Host kernel (it should be the default).

6. Install the Dialogic® System Release 6.1 for Linux Service Update. Remove `ct_intel` from executing on boot: `chkconfig ct_intel off`.

7. Test Switchover before making the system live (mainly check the Domain Status).

- Notes:**
1. If Dialogic® Services (build 171 or older) were started on the STANDBY, then the System Monitor would be checking the board status. Since the driver does have the PCI bus, it cannot send board status messages. Therefore, the System Monitor will think it has a crashed board. The Fault Detector tries to get a SRAM dump (which come back w/ 0). This will go on every few minutes.
 2. In a glare condition, if a Switchover is happening at the same time the Fault Detector is running, a panic can occur since the SRAM is not properly activated.
 3. With build 181, the Fault Detector is disabled. The System Monitor will still do board status but won't inform the Fault Detector to start. This should prevent the Fault Detector from crashing. But there still could be a Hot Swap related crash or SBC hang due to some hardware error.

3.2.2 Dialogic® DM3 Architecture Products on Linux Configuration Guide

Updates to **Section 2.3, Media Loads**

Because of features introduced in the Service Update, several new media loads are available for the Dialogic® DM3 Boards. These media loads should be documented in **Section 2.3**. For information about these media loads, see [Section 1.8, “New Media Loads for Dialogic® DM/V2400A-cPCI and DMV4800BC Media Boards”](#), on page 44, [Section 1.18, “New Media Load for Dialogic® DMV1200BTEP Media Boards”](#), on page 60, [Section 1.46, “New Media Load for Dialogic® DMV600BTEC Boards”](#), on page 96, [Section 1.58, “New Media Loads for Dialogic® DMV1200BTEC Boards”](#), on page 104, and [Section 1.66, “New Media Load for Dialogic® DMV4800BC Boards”](#), on page 111 of this Release Update.

Update to **Section 2.3.1.1, DM/IP, DM/V, DM/V-A, and DM/V-B Boards**

The following information should be added to indicate that some coders are not supported on the **Dialogic® DM/V4800BC Board** with certain media loads:

When using the Dialogic® DM/V4800BC Board, the maximum bit rate for standard play/record with any basic media load type (for example, media load 1B) is 64 Kbps. The following coders, which are normally part of the standard basic voice media load,

are not supported on the DM/V4800BC Board with any basic media load due to density considerations:

- Linear PCM, 8 KHz sampling rate, 16-bit resolution (128 Kbps) VOX and WAVE
- Linear PCM, 11 KHz sampling rate, 8-bit resolution (88 Kbps) VOX and WAVE
- Linear PCM, 11 KHz sampling rate, 16-bit resolution (176 Kbps) VOX and WAVE

Updates to **Section 2.3.1.1, DM/IP, DM/V, DM/V-A, and DM/V-B Boards**

The first paragraph in **Section 2.3.1.1** should read:

- Dialogic® DM/V, DM/V-A, and DM/V-B Media Boards are supported by media loads 1 through 10 and universal media loads 1 through 4.

In **Table 3, Channel Densities by Board and Media Load (Universal)**, the table footnote about echo cancellation should be changed as follows:

- Default configuration is EEC (enhanced EC, 32 ms) for CSP supported ML, unless otherwise indicated or set in the component named [0x2c] in the respective CONFIG file. You can only change it to a lower EC tail length, by changing the CSP parameter 0x2c03 accordingly in the respective CONFIG file. Conferencing EC, however, will always be 16 ms, regardless of the EC parameter setting.

Update to **Section 2.3.1.2, DM3 Analog Voice Boards**

The second paragraph in **Section 2.3.1.2** should read:

- Refer to Table 4 for a list of channel densities.

Update to **Section 2.5, CT Bus Clock Fallback (IPY00036875)**

The following information about clock fallback in mixed systems (systems that have different board technologies such as Dialogic® DM3 Boards, Dialogic® Springware Boards, and third party boards) should be added:

Clock Fallback in Mixed Systems

In mixed systems with different board technologies (e.g., Dialogic® DM3 Boards, Dialogic® Springware Boards, and third party boards), all boards used as clock masters and reference masters should be the same technology. The following table shows the valid clocking scenarios for mixed systems.

Valid Clocking Scenarios for Mixed Systems								
Parameter Settings						Resulting Clock Fallback Setup		
Primary Clock Master	Secondary Clock Master	Primary Reference Master	Secondary Reference Master	Clocking Daemon Mode	Network Reference Source	Clock Master Fallback	Reference Master Fallback	CT Clock Source (Resolved)
DM3	DM3	DM3	DM3	Active	DM3	Yes	Yes	DM3, dynamic
Springware	Not applicable	Springware	Not applicable	Passive	Springware	No	No	Springware, static as set
Third party	Third party	Third party	Third party	Passive	Third party	Third party dependent	Third party dependent	Third party

Any other configuration can lead to unpredictable CT Bus clocking behavior or PSTN network alarms. For example, it is **not** possible to set a Dialogic® DM3 Board as the primary clock master with a Dialogic® Springware Board as a reference master. The following table shows

an example of an **invalid** clocking scenario, where one would expect the primary clock master to derive its clock from the NETREF; however, it will default to internal oscillator due to initialization timing between the mixed technologies.

Example of an Invalid Clocking Scenario for Mixed Systems								
Parameter Settings						Resulting Clock Fallback Setup		
Primary Clock Master	Secondary Clock Master	Primary Reference Master	Secondary Reference Master	Clocking Daemon Mode	Network Reference Source	Clock Master Fallback	Reference Master Fallback	CT Clock Source (Resolved)
DM3	DM3	Springware	Don't care	Active	Springware	Yes	No	DM3, internal oscillator

Update to **Section 3.4, [NFAS] Section**

The third note about NFAS D channel backup (DCBU) supported only on ISDN NI-2 protocol is incorrect. DCBU is supported on 4ESS, 5ESS, and NI-2.

Update to **Section 4.5.2, Define the Trunk Configuration**

Step 6 in **Section 4.5.2** has been revised as follows:

6. From the Trunk Configuration - Specify Media Load screen, select a media load by typing the number corresponding to the media load you wish to assign to this board and then press Enter.

The Trunk Configuration - Specify Protocols for the Trunks screen will then be displayed, Figure 15 or Figure 16.

Figure 15. Trunk Configuration - Specify Protocols for the Trunks Screen (DM/V-B)

Trunk Configuration - Specify Protocols for the Trunks

Select trunk protocols from the following list. Refer to a protocol by its Protocol ID. All the selected protocols must have the same Group number.

Number of trunks for this board: 4

Enter the trunk number(s) for the board and the protocol(s) you want to select for each trunk in the following format:

<Trunk Range>:<Protocol ID>,<Trunk Range>:<Protocol ID> etc.

ProtocolID	ProtocolName	Group	ProtocolID	ProtocolName	Group
------------	--------------	-------	------------	--------------	-------

1	4ESS (T1)	1	2	5ESS (T1)	1
3	NTT (T1)	1	4	NI2 (T1)	1
5	DMS (T1)	1	6	QSIGT1 (T1)	1
7	QSIGEl (E1)	1	8	NET5 (E1)	1
9	T1CC (T1)	1	10	CAS (T1)	1
11	E1CC (E1)	1	12	R2MF (E1)	1
13	DPNSS (E1)	2	14	DASS2 (E1)	2

(s to save, x to save & quit, q to quit) the configuration
p to return to Trunk Configuration - Specify Media Load
r to return to Modify Board Settings
? for help and ! for navigation help
Enter the Trunk Range and Protocol ID numbers: 1-2:9,3-4:10

Figure 16. Trunk Configuration - Specify Protocols for the Trunks Screen (DMN160TEC or DMT160TEC)

Trunk Configuration - Specify Protocols for the Trunks

Select trunk protocols from the following list. Refer to a protocol by its Protocol ID. All the selected protocols must have the same Group number.

Number of trunks for this board: 4

Enter the trunk number(s) for the board and the protocol(s) you want to select for each trunk in the following format:

<Trunk Range>:<Protocol ID>,<Trunk Range>:<Protocol ID> etc.

ProtocolID	ProtocolName	Group		ProtocolID	ProtocolName	Group
1	4ESS (T1)	1		2	5ESS (T1)	1
3	NTT (T1)	1		4	NI2 (T1)	1
5	DMS (T1)	1		6	QSIGT1 (T1)	1
7	ISDNT1CC (T1)	1		8	QSIGE1 (E1)	1
9	NET5 (E1)	1		10	ISDNE1CC (E1)	1
11	T1CC (T1)	1		12	CAS (T1)	1
13	E1CC (E1)	1		14	R2MF (E1)	2

(s to save, x to save & quit, q to quit) the configuration
p to return to Trunk Configuration - Specify Media Load
r to return to Modify Board Settings
? for help and ! for navigation help
Enter the Trunk Range and Protocol ID numbers: 1-2:9,3-4:10

The Trunk Configuration - Specify Protocols for the Trunks screen allows you to choose a protocol for each trunk on a Dialogic® DM/V-B Board that has network interfaces, a Dialogic® DMN160TEC Board, or a Dialogic® DMT160TEC Board. You may assign the same protocol or different protocols to each trunk on the board, but all of the protocols must belong to the same group number.

- Notes:**
1. The Dialogic® DMN160TEC Board does not support CAS or R2MF.
 2. When assigning different protocols to trunks on a Dialogic® DMT160TEC Board, in which one or more trunks includes the E1CC protocol, the following considerations apply:
 - A combination of ISDN, E1CC, and R2MF protocols is not supported.
 - A combination of ISDN and E1CC (CC using PDK) is not supported. Instead, use ISDN and ISDNE1CC.
 - A combination of R2MF and ISDNE1CC (CC using ISDN) is not supported. Instead, use R2MF and E1CC.

Update to **Section 4.6.3, Perform Advanced TDM Bus Configuration**

The information following completion of the board and TDM bus configuration in **Section 4.6.3** should read as follows:

```
Would you like to configure SNMP on this system (y/n, default=n)?
```

If you installed the SNMP agent software, type *y*; otherwise type *n*.

Notes: 1. Do not enter *y* to configure SNMP if you have not installed the SNMP agent software. If you do, the configuration procedure is aborted and you will be prompted to run the installation script (*install.sh*) again so you can install the SNMP agent software.

2. If you installed SNMP agent software, but enter *n*, the following messages will appear:

```
WARNING:
```

```
Configuration of the SNMP is incomplete.
```

```
Please run '/usr/dialogic//bin/dlgcsnmpconf' to configure the SNMP.
```

3. If you installed SNMP agent software and enter *y* to configure SNMP, the SNMP Agents Configuration Tool is automatically invoked when board configuration is complete.

Update to **Section 4.8, Completing the Configuration Utility (config.sh)**

The information in **Section 4.8** should read as follows:

When *config.sh* is complete, the following messages are displayed:

```
Configuration is now complete.
```

```
Press <Enter> to exit the configuration tool.
```

Before using the software, you must ensure that the Intel(R) Dialogic(R) environment variables are set using any one of the following actions:

(a) At a BASH shell prompt execute: `source /etc/profile.d/ct_intel.sh`

(b) At a C shell prompt execute: `source /etc/profile.d/ct_intel.csh`

(c) Logout and login

(d) Reboot system

The Intel(R) Dialogic(R) system services will automatically start every time the system is rebooted. To start and stop system services manually, use the *dlstart* and *dlstop* scripts found in `/usr/dialogic//bin`.

Notice for CompactPCI users:

=====

If you wish to use the Hot Swap Kit or Redundant Host software, and it is not already installed on your system, please change into the PPS directory and run the '*hsk_install.sh*' script. Once this script has completed, a system reboot is required.

Note: If SNMP agent software is installed, but not configured, the following message will be displayed following the *dlstart* command:

WARNING:

Configuration of the SNMP is incomplete.

Please run 'usr/dialogic//bin/dlgcsnmpconf' to configure SNMP.

Update to **Section 6.8, [0x3b] Parameters**

Because of a feature introduced in the Service Update, a new parameter can be added to the [0x3b] section of the CONFIG file to adjust the AGC Noise Level Lower Threshold. For information about this parameter, see [Section 1.66, “New Media Load for Dialogic® DMV4800BC Boards”](#), on page 111 of this Release Update.

Update to **Section 6.8, [0x3b] Parameters**

Information about parameters **0x3b03** and **0x3b04** should be added to this section as follows:

Note: This information is intended for experienced users of the DM3 conferencing feature. Changing the default parameter settings is **not** recommended, as this could introduce negative effects on the audio quality and conferencing experience of the participants.

CSUMS_ActTalkerPartiesMinNum

Number: 0x3b03

Description: Specifies the number of talkers in a conference before Active Talker mode is enabled.

Note: Conference Active Talker mode, though related, should not be confused with the Active Talker detection feature.

Values: 0 [default] to 0xff (255).

Guidelines: Refer to the guidelines for the **CSUMS_SmartScalingPartiesMinNum** parameter below.

CSUMS_SmartScalingPartiesMinNum

Number: 0x3b04

Description: Specifies the number of talkers in a conference before scaling mode is enabled.

Values: 0 [default] to 0xff (255).

Guidelines: Audio conferencing provides a mechanism for audio summation of two or more parties in a conference. There are three possible summing modes, which are controlled by CSUMS parameters **0x3b03** and **0x3b04** in the configuration file.

By default, both active talker and scaling are enabled. When parameters **0x3b03** and **0x3b04** are both set to their default values of 0, the default summing mode is Active Talker Summation mode, which sums the three loudest parties. This is advantageous in **large conferences**. Since only the three loudest parties are summed, background noise is reduced. However, there may be times with **small conferences** when a different summation mode is preferable, for example, with soft speakers or when the energy is too low (as with analog lines). The other summation modes are:

- Smart scaling mode - the summation of all parties, but scaling is only done on the ones who are talking.
- No scaling - pure summation, can be used if you want full voice energy in the conference.

The settings for parameters **0x3b03** and **0x3b04** determine the summing mode as shown in the following table.

Parameter 0x3b03, CSUMS_ActTalkerPartiesMinNum	Parameter 0x3b04, CSUMS_SmartScalingPartiesMinNum	Summing Mode
0 (default)	0 (default)	Active Talker Detection (default)
> Conf_MaxTotalParties	0 (default)	Smart Scaling
> Conf_MaxTotalParties	> Conf_MaxTotalParties	No Scaling
Conf_MaxTotalParties is the setting for parameter 0x3926 in the configuration file, e.g., SetParm=0x3926,120 !Conf_MaxTotalParties		

To disable the Active Talker algorithm, set the parameter **0x3b03** to a value larger than the maximum number of conferences per DSP; setting it to **Conf_MaxTotalParties**, or per board total number of parties, will suffice, to a maximum of 255.

Even without Active Talker, scaling is also enabled by default. If not required, set the parameter **0x3b04** to a number larger than the maximum number of parties per DSP, and again using **Conf_MaxTotalParties** will suffice, to a maximum of 255.

Update to **Section 6.10, [lineAdmin.x] Parameters (Digital Voice)**

In the description of the **SignalingType** parameter, the note about NFAS D channel backup (DCBU) supported only on ISDN NI-2 protocol is incorrect. DCBU is supported on 4ESS, 5ESS, and NI-2.

Update to **Section 6.13, [NFAS.x] Parameters**

In the description of the **NFAS_Standby_IntlID** parameter, the note about NFAS D channel backup (DCBU) supported only on ISDN NI-2 protocol is incorrect. DCBU is supported on 4ESS, 5ESS, and NI-2.

Update to **Section 6.22, [CHP] ISDN Protocol Variant Definitions (IPY00045267)**

The values shown for the **ProtocolType** parameter are incorrect. The correct values for the **ProtocolType** parameter are:

- 1: 4ESS
- 2: 5ESS
- 3: DMS100 and DMS250
- 4: NTT
- 7: NET5
- 8: DASS2
- 9: DPNSS
- 10: QSIG1
- 11: QSIGT1
- 12: NI2

3.2.3 Dialogic® IPT Series Products on Linux Configuration Guide

There are currently no updates to this document.

3.2.4 Dialogic® Springware Architecture Products on Linux Configuration Guide

Update to **Section 3.6.2, Defining TDM Bus Role Settings** (IPY00036875)

The following information about clock fallback in mixed systems (systems that have different board technologies such as Dialogic® DM3 Boards, Dialogic® Springware Boards, and third party boards) should be added:

Clock Fallback in Mixed Systems

In mixed systems with different board technologies (e.g., Dialogic® DM3 Boards, Dialogic® Springware Boards, and third party boards), all boards used as clock masters and reference masters should be the same technology. The following table shows the valid clocking scenarios for mixed systems.

Valid Clocking Scenarios for Mixed Systems								
Parameter Settings						Resulting Clock Fallback Setup		
Primary Clock Master	Secondary Clock Master	Primary Reference Master	Secondary Reference Master	Clocking Daemon Mode	Network Reference Source	Clock Master Fallback	Reference Master Fallback	CT Clock Source (Resolved)
DM3	DM3	DM3	DM3	Active	DM3	Yes	Yes	DM3, dynamic
Springware	Not applicable	Springware	Not applicable	Passive	Springware	No	No	Springware, static as set
Third party	Third party	Third party	Third party	Passive	Third party	Third party dependent	Third party dependent	Third party

Any other configuration can lead to unpredictable CT Bus clocking behavior or PSTN network alarms. For example, it is **not** possible to set a Dialogic® DM3 Board as the primary clock master with a Dialogic® Springware Board as a reference master. The following table shows an example of an **invalid** clocking scenario, where one would expect the primary clock master to derive its clock from the NETREF; however, it will default to internal oscillator due to initialization timing between the mixed technologies.

Example of an Invalid Clocking Scenario for Mixed Systems								
Parameter Settings						Resulting Clock Fallback Setup		
Primary Clock Master	Secondary Clock Master	Primary Reference Master	Secondary Reference Master	Clocking Daemon Mode	Network Reference Source	Clock Master Fallback	Reference Master Fallback	CT Clock Source (Resolved)
DM3	DM3	Springware	Don't care	Active	Springware	Yes	No	DM3, internal oscillator

Update to **Section 3.6.3, Perform Advanced TDM Bus Configuration**

The information following completion of the board and TDM bus configuration in **Section 3.6.3** should read as follows:

Would you like to configure SNMP on this system (y/n, default=n)?

If you installed the SNMP agent software, type y; otherwise type n.

- Notes:**
1. Do not enter y to configure SNMP if you have not installed the SNMP agent software. If you do, the configuration procedure is aborted and you will be prompted to run the installation script (install.sh) again so you can install the SNMP agent software.
 2. If you installed SNMP agent software, but enter n, the following messages will appear:

WARNING:

Configuration of the SNMP is incomplete.

Please run '/usr/dialogic//bin/dlgcsnmpconf' to configure the SNMP.

3. If you installed SNMP agent software and enter y to configure SNMP, the SNMP Agents Configuration Tool is automatically invoked when board configuration is complete.

Update to **Section 3.8, Completing the Configuration Utility (config.sh)**

The information in **Section 3.8** should read as follows:

When *config.sh* is complete, the following messages are displayed:

Configuration is now complete.

Press <Enter> to exit the configuration tool.

Before using the software, you must ensure that the Intel(R) Dialogic(R) environment variables are set using any one of the following actions:

- (a) At a BASH shell prompt execute: source /etc/profile.d/ct_intel.sh
- (b) At a C shell prompt execute: source /etc/profile.d/ct_intel.csh
- (c) Logout and login
- (d) Reboot system

The Intel(R) Dialogic(R) system services will automatically start every time the system is rebooted. To start and stop system services manually, use the dlstart and dlstop scripts found in /usr/dialogic//bin.

Notice for CompactPCI users:

=====

If you wish to use the Hot Swap Kit or Redundant Host software, and it is not already installed on your system, please change into the PPS directory and run the 'hsk_install.sh' script. Once this script has completed, a system reboot is required.

- Note:** If SNMP agent software is installed, but not configured, the following message will be displayed following the dlstart command:

WARNING:

Configuration of the SNMP is incomplete.

Please run '/usr/dialogic//bin/dlgcsnmpconf' to configure SNMP.

Update to **Section 3.11, Configuring Voice Parameters**

Because of a feature introduced in the Service Update, two new parameters can be added to the *voice.prm* file to set the firmware play and record buffer size. For information about these parameters, see [Section 1.17, “Setting Data Transfer Buffer Size below 1K for Dialogic® Springware Boards”](#), on page 57 of this Release Update.

Update to **Chapter 4, Dialogic.Cfg Parameter Reference** for **CSPEXtraTimeSlot** parameter (IPY00041018)

The guidelines for **CSPEXtraTimeSlot** should be changed to the following:

For Dialogic® Springware Boards that use CSP, you must set the **CSPEXtraTimeSlot** and **EC_Resource** parameters to ON.

Update to **Chapter 4, Dialogic.Cfg Parameter Reference** for **EC_Resource** parameter (IPY00041018)

The following guideline for **EC_Resource** should be added:

For Dialogic® Springware Boards that use CSP, you must set the **EC_Resource** and **CSPEXtraTimeSlot** parameters to ON.

Update to **Chapter 4, Dialogic.Cfg Parameter Reference** for **FirmwareFile** parameter

The section about the **FirmwareFile** parameter should be changed to the following:

FirmwareFile

Usage: Board parameter, optional, applies to all baseboards.

Description: Specifies the name of a firmware load file for the system software to download to the board. This firmware file takes the place of the file that is normally downloaded.

For specifying the firmware load file of the second span on boards that have two spans, use the **FirmwareFile2** parameter.

Guidelines: When you execute Genload, the file that you specify here is located according to the following sequence:

- If a full pathname is specified (for example, **FirmwareFile = /usr/dialogic/data/spandti.fwl**), that file is used.
- If only a file name is specified (for example, **FirmwareFile = spandti.fwl**) and the file is in the directory from which Genload is executed, that file is used.
- Otherwise, the default firmware file location is */usr/dialogic/data*.

The default firmware file is the file specified using the **ISDNProtocol** parameter. If the **ISDNProtocol** parameter is set to NONE, the *spanplus.fwl* file is downloaded.

For Dialogic® Springware Boards that support Dialogic® Continuous Speech Processing (CSP), a special firmware file is required. To enable CSP capability for Dialogic® Springware Boards, you must explicitly specify the CSP firmware file. See the **FirmwareFile2** parameter for a list of standard (default) and CSP-specific firmware files.

For Dialogic® Springware Boards that support fax, a special firmware file is required. To enable DSP-fax capability for Dialogic® Springware Boards, you must explicitly specify the fax firmware

file. See the **FirmwareFile2** parameter for a list of standard (default) and fax-specific firmware files.

Note: DSP-based fax and CSP cannot be used together on the same board.

Values: The firmware load files are installed in */usr/dialogic/data* and most have the extension *.fwl*.

Default value: Without this parameter, Genload automatically selects the correct firmware file to download.

Update to **Chapter 4, Dialogic.Cfg Parameter Reference** for **FirmwareFile2** parameter
The section about the **FirmwareFile2** parameter should be changed to the following:

FirmwareFile2

Usage: Board parameter, optional, applies to boards with two spans (for example, Dialogic® D/480JCT-2T1) and to enable Dialogic® Continuous Speech Processing (CSP) capability or DSP-based fax on Dialogic® Springware Boards that support these features.

Note: DSP-based fax and CSP cannot be used together on the same board.

Description: Specifies the name of a firmware load file for the system software to download to the second span of an applicable board. This firmware file takes the place of the file that is normally downloaded.

Specify the firmware load file for the first span using the **FirmwareFile** parameter.

Guidelines: For Dialogic® Springware Boards that support CSP or DSP-based fax, a special firmware file is required. To enable CSP or fax capability for Dialogic® Springware Boards, you must explicitly specify the CSP or fax firmware file.

For Dialogic® D/480JCT-1T1 and D/600JCT-1E1 Boards, you can provide for ISDN support on one span and CSP or fax support on the other by using two separate firmware files, one for each span.

- On the first span, you can specify an ISDN protocol and then the specific firmware file required for that ISDN protocol will be automatically downloaded to the board for that span. CSP capability is not available on this span.
- On the second span, you can enable CSP or fax capability, without ISDN support, by specifying the CSP or fax firmware file for that span and setting the ISDN protocol parameter value to **none**.

NOTE: For E1 and T1 boards that support CSP or fax, specifying both an ISDN protocol (with **ISDNProtocol** or **ISDNProtocol2** parameter) and a CSP or fax firmware file (with **FirmwareFile** or **FirmwareFile2** parameter) for the same span results in a download failure to that span. The Dialogic® System Service will not start.

Table 3 summarizes CSP and ISDN interoperability for Dialogic® D/480JCT-1T1 and D/600JCT-1E1 Boards. Table 4 summarizes fax and ISDN interoperability for Dialogic® D/480JCT-1T1 and D/600JCT-1E1 Boards.

Table 3. CSP and ISDN Interoperability for Dialogic® D/480JCT-1T1 and D/600JCT-1E1 Boards

D/480JCT-1T1 or D/600JCT-1E1	ISDN Protocol Setting	Firmware File Setting	Result
First span	None	Standard firmware file	First span does not support ISDN.
	Specific ISDN protocol selected using the ISDNProtocol parameter	Firmware file specific to ISDNProtocol parameter automatically downloaded	First span supports ISDN.
Second span	None	CSP firmware file	Second span supports CSP.

Table 4. Fax and ISDN Interoperability for Dialogic® D/480JCT-1T1 and D/600JCT-1E1 Boards

D/480JCT-1T1 or D/600JCT-1E1	ISDN Protocol Setting	Firmware File Setting	Result
First span	None	Standard firmware file	First span does not support ISDN.
	Specific ISDN protocol selected using the ISDNProtocol parameter	Firmware file specific to ISDNProtocol parameter automatically downloaded	First span supports ISDN.
Second span	None	Fax firmware file	Second span supports fax.

For Dialogic® **D/480JCT-2T1** Boards, you can provide for CSP support on one span and ISDN support on the other as follows:

- For CSP on the first span and ISDN on the second span:

```
[Genload - PCI ID xx]
FirmwareFile=spcsp.fwl /*for CSP on first span*/
ISDNProtocol2=DMS /*or other ISDN protocol for ISDN on second span*/
```

- For ISDN on the first span and CSP on the second span; note that the **ISDNProtocol2** parameter must explicitly be set to **none** in this case:

```
[Genload - PCI ID xx]
#FirmwareFile= (no FirmwareFile specified for first span)
ISDNProtocol=DMS /*or other ISDN protocol for ISDN on first span*/
FirmwareFile2=spcsp.fwl /*for CSP on second span*/
ISDNProtocol2=NONE /*must be set to none or else Genload will try to force
                    ISDNProtocol value to ISDNProtocol2 and this is not
                    supported with CSP*/
```

Values: Table 5 lists both the standard (default) firmware files and the CSP firmware files for Dialogic® Springware Boards that support the CSP feature. Table 6 lists both the standard (default) firmware files and the DSP-based fax firmware files for Dialogic® Springware Boards that support the fax feature.

Table 5. Firmware Files for Default and CSP Configurations

Board Type	Standard (Default) Configuration		CSP Configuration	
	Firmware File	Firmware File2	Firmware File	Firmware File2
D/120JCT-LS	<i>spanplus.fwl</i>	not applicable	<i>spcsp.fwl</i>	not applicable
D/240JCT-T1	<i>spanplus.fwl</i>	not applicable	<i>spcsp.fwl</i>	not applicable
D/480JCT-2T1	<i>spanplus.fwl</i> or <i>ISDNProtocol</i> parameter value	<i>spanplus.fwl</i> or <i>ISDNProtocol2</i> parameter value	<i>spcsp.fwl</i>	<i>spcsp.fwl</i>
D/480JCT-1T1	<i>spanplus.fwl</i> or <i>ISDNProtocol</i> parameter value	<i>spanplus.fwl</i>	<i>spanplus.fwl</i> or <i>ISDNProtocol</i> parameter value	<i>spcsp.fwl</i>
D/600JCT-1E1	<i>spanplus.fwl</i> or <i>ISDNProtocol</i> parameter value	<i>spanplus.fwl</i>	<i>spanplus.fwl</i> or <i>ISDNProtocol</i> parameter value	<i>spe1csp.fwl</i>

Table 6. Firmware Files for Default and Fax Configurations

Board Type	Standard (Default) Configuration		CSP Configuration	
	Firmware File	Firmware File2	Firmware File	Firmware File2
D/120JCT-LS	<i>spanplus.fwl</i>	not applicable	<i>spcsp.fwl</i>	not applicable
D/240JCT-T1	<i>spanplus.fwl</i>	not applicable	<i>spcsp.fwl</i>	not applicable
D/480JCT-2T1	<i>spanplus.fwl</i> or <i>ISDNProtocol</i> parameter value	<i>spanplus.fwl</i> or <i>ISDNProtocol2</i> parameter value	<i>spcsp.fwl</i>	<i>spfax.fwl</i>
D/480JCT-1T1	<i>spanplus.fwl</i> or <i>ISDNProtocol</i> parameter value	<i>spanplus.fwl</i>	<i>spanplus.fwl</i> or <i>ISDNProtocol</i> parameter value	<i>spfax.fwl</i>
D/600JCT-1E1	<i>spanplus.fwl</i> or <i>ISDNProtocol</i> parameter value	<i>spanplus.fwl</i>	<i>spanplus.fwl</i> or <i>ISDNProtocol</i> parameter value	<i>spe1fax.fwl</i>

Default value: See Table 5 or Table 6.

Update to **Chapter 4, Dialogic.Cfg Parameter Reference** for **ISDNProtocol** parameter
The section about the **ISDNProtocol** parameter should be changed to the following:

ISDNProtocol

Usage: Global or board parameter, optional, applies to boards with a digital network interface.

Description: Specifies that the board's digital network interface should be configured for ISDN using the selected ISDN protocol.

For specifying the ISDN protocol of the second span on boards that have two spans, use the **ISDNProtocol2** parameter.

Guidelines: The ISDN PRI Protocols package (DLGCpri) is installed with the Dialogic® Springware Software.

If you use the **ISDNProtocol** parameter to download an ISDN protocol firmware file to a board, the **FirmwareFile** parameter must use its default value.

Note: For E1 and T1 boards that support Dialogic® Continuous Speech Processing (CSP), specifying an ISDN protocol and a CSP firmware file for the same span results in a download failure to that span. The Dialogic® System Service will not start.

Note: For E1 and T1 boards that support DSP-based fax, specifying an ISDN protocol and a fax firmware file for the same span results in a download failure to that span. The Dialogic® System Service will not start.

For additional information about CSP and DSP-based fax interaction with ISDN operation, see the [FirmwareFile2](#) parameter in this section.

Values: Valid values for **ISDNProtocol** are:

NONE	No ISDN protocol is used
4ESS	AT&T 4ESS custom switch TR41449/TR41459
5ESS	AT&T 5ESS custom switch 505-900-322
CTR4	EURO-ISDN ETSI300-102
DASS2	British National BTNR-190-1985
DMS	Northern Telecom custom switch A211-1 and A211-4
DPNSS (separately ordered)	British Private Branch Exchange DASS2 extension
ETN	EURO-ISDN ETSI300-102 for T-1
ETU	EURO-ISDN ETSI300-102 for T-1
NE1	EURO-ISDN ETSI300-102
NI2	National ISDN-2 Bellcore Special Report SR-NWT-002343

NT1	T-1 Network Emulation TR41449/TR41459
NTT	Japanese National ISDN INS-Net 1500
QNT	Q.SIG ISO 11572, ISO 11574
QTE	Q.SIG ISO 11572, ISO 11574
QTN	Q.SIG ECMA-142/143 for T-1
QTU	Q.SIG ECMA-142/143 for T-1
VNNT	French National ISDN VN3 (Network Termination)

Default value: NONE (no ISDN protocol is used)

Update to **Chapter 4, Dialogic.Cfg Parameter Reference** for **ISDNProtocol2** parameter
The section about the **ISDNProtocol2** parameter should be changed to the following:

ISDNProtocol2

Usage: Board parameter, optional, applies to digital network interface boards with two spans (for example, Dialogic® D/480JCT-2T1).

Description: Specifies that the board's second digital network interface should be configured for ISDN using the selected ISDN protocol.

Specify the ISDN protocol for the first span using the **ISDNProtocol** parameter (which may be a global and/or board parameter).

Guidelines: The ISDN PRI Protocols package (DLGCpri) is installed with the Dialogic® Springware Software.

If you use the **ISDNProtocol2** parameter to download an ISDN protocol firmware file, the **FirmwareFile2** parameter must use its default value.

Note: For E1 and T1 boards that support Dialogic® Continuous Speech Processing (CSP), specifying an ISDN protocol and a CSP firmware file for the same span results in a download failure to that span. The Dialogic® System Service will not start.

Note: For E1 and T1 boards that support DSP-based fax, specifying an ISDN protocol and a fax firmware file for the same span results in a download failure to that span. The Dialogic® System Service will not start.

For additional information about CSP and DSP-based fax interaction with ISDN operation, see the [FirmwareFile2](#) parameter in this section.

Values: See the [ISDNProtocol](#) parameter in this section.

Default value: NONE (no ISDN protocol is used)

3.2.5 Dialogic® Global Call Country Dependent Parameters (CDP) for PDK Protocols Configuration Guide

Update to **Chapter 10, Brazil R2 Bidirectional Protocol Parameter Configuration**

The **CDP_SKIP_A3_AND_A4_PULSE** parameter should be added to this chapter as follows:

CDP_SKIP_A3_AND_A4_PULSE (Inbound)

Description: Specifies when to send ANI after DNIS when overlap sending is enabled and the Append F flag is disabled. This parameter is valid only if **CDP_OVERLAP_SENDING_ENABLED=1** and **CDP_FLAG_APPEND_F=0**.

Values:

- 0 [default]: Protocol waits for A3 or A4 pulse and then requests Category before requesting ANI. This is the default behavior when **CDP_OVERLAP_SENDING_ENABLED=1** and **CDP_FLAG_APPEND_F=0**.
- 1: Protocol requests ANI immediately after DNIS, without waiting for A3 or A4.

Update to **Chapter 10, Brazil R2 Bidirectional Protocol Parameter Configuration**

The “Values” section for the **CDP_FLAG_APPEND_F** parameter should be updated to refer to the **CDP_SKIP_A3_AND_A4_PULSE** parameter as follows:

Values:

- 0 [default]: No tone will be sent to the remote end. In this case, A3 or A4 pulse is expected to be received from the remote end. In a case of overlapped sending (see description of **CDP_OVERLAP_SENDING_ENABLED** parameter), the remote end may also send A1 to request more information.
To skip the A3 or A4 pulse and send ANI immediately after DNIS, set **CDP_SKIP_A3_AND_A4_PULSE=1**.
- 1: ‘f’ (I-15) will be sent to the remote end, indicating the end of information.

3.3 OA&M Documentation Updates

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

- [Dialogic® System Release 6.1 for Linux Administration Guide](#)
- [Dialogic® SNMP Agent Software for Linux Administration Guide](#)
- [Dialogic® System Software Diagnostics Guide](#)
- [Dialogic® Board Management API Library Reference](#)
- [Dialogic® OA&M API Programming Guide](#)
- [Dialogic® OA&M API Library Reference](#)

3.3.1 Dialogic® System Release 6.1 for Linux Administration Guide

Update to **Section 4.2, QScript Utilities**

Tcl/Tk no longer needs to be installed. Also, no symbolic links are required to simulate Tcl/Tk 8.3. The latest QScript delivers libraries (libqscript & libqscriptgx) that are used instead of Tcl/Tk.

Therefore the following changes should be made:

In **Section 4.2, QScript Utilities**, remove the following text from the first paragraph:

- and is implemented using the Tcl/Tk generic scripting language. The QScript utilities require the Tcl/Tk GUI environment.

In **Section 4.2, QScript Utilities**, remove **Section 4.2.1.3, Requirements for tcl**.

Update to **Section 4.7, IPT Series Firmware Update (iptfwupdate)** (PTR# 36343)

The procedure for using the IPT Series Firmware Update utility has been revised. The new procedure is included below:

The following procedure provides instructions for using the *IPT Series Firmware Update* utility:

1. Prior to installing the update, check and record the version number of the current firmware file on the system by using the Getver diagnostic utility from the */data* directory:

```
Getver pmac_stl.bin
```

An output similar to the following is displayed:

```
Version: 1.10   Build: NBD_1.10.10_B4
```

Note: The value returned is the version number of the firmware file and not the version number on the board.

Note: You may want to copy the current firmware file to a safe location as a backup.

For more information on the Getver utility, see the *Dialogic® System Software Diagnostics Guide*.

2. Obtain the new firmware file for Dialogic® IPT Boards, which is located in the */data* directory under the Dialogic® System Release 6.1 for Linux installation directory following standard installation. This firmware file is typically installed as part of a system release or a service update.
3. After installing the update, check the version number of the firmware file on the system by using the Getver diagnostic utility from the */data* directory:

```
Getver pmac_stl.bin
```

If the firmware file obtained in step 3 is newer than the one recorded in step 1, continue to step 4. If the firmware file obtained in step 3 is the same as the one recorded in step 1, stop here; you do not need to perform the firmware update.

4. Optional. To determine the board number if it is not already known, invoke the *IPT Series Administration* utility. The board number is used as input to -b parameter in step 7.

```
pmacadmin -l
```

5. Check the current version number of the firmware file on the board as follows:

5a. Open *pmac.cfg* located in the */cfg* directory.

5b. Look for the version number, a hexadecimal value, following the *IPTBoardSoftwareVersion* parameter. For example:

```
IPTBoardSoftwareVersion = 0x010a0a04
```

This hexadecimal value corresponds to the decimal value returned by *Getver* in step 1 (1.10.10.4).

6. Reset all boards in the system and stop system services by issuing *dlstop*.

7. Invoke the *IPT Series Firmware Update* utility to download the firmware to the board. For example:

```
iptfwupdate -f $INTEL_DIALOGIC_DIR/data/pmac_stl.bin -b 0 -t 1
```

8. Restart all boards in the system as well as system services by issuing *dlstart*.

9. Check the version number of the new firmware file on the board as follows:

9a. Open *pmac.cfg* located in the */cfg* directory.

9b. Look for the version number, a hexadecimal value, following the *IPTBoardSoftwareVersion* parameter. This hexadecimal value should correspond to the decimal value returned by *Getver* in step 3.

Update to **Section 4.9, List Board Information Utility (listboards)** (PTR# 36306)

The following note at the beginning of Section 4.9 should be **deleted**, because there is no longer a dependency on installing the *DLGCdmdev* RPM in order to use *listboards* in a Springware-only system:

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 an Intel NetStructure menu item during installation of the software release (see the *Software Installation Guide*).

Update to **Section 4.9, List Board Information Utility (listboards)** (IPY00045020)

A note should be added following **Figure 7, Listboards Level 1 Example Output**:

Note: In *listboards* level 1 output, the “Slot” column shows the slot number for cPCI boards. For PCI boards, it shows the thumbwheel ID.

Update to **Section 4.16, System Logging**

The System Logging facility has been integrated with Runtime Trace Facility (RTF). The environment *DLG_TRACE_LEVEL* is replaced by the RTF module *OAMSYSLOG*. The *dlgsyslogger.log* and *oam.log* files are no longer generated and have now been replaced by RTF log files. Refer to the “Runtime Trace Facility (RTF) Reference” chapter in the *Dialogic® System Software Diagnostics Guide* for more details on RTF logging.

3.3.2 Dialogic® SNMP Agent Software for Linux Administration Guide

Update to **Chapter 1, Administration Overview**

A new section titled **Section 1.3, SNMP Related Startup Scripts** should be added to Chapter 1 with the following content:

This section provides information about SNMP related startup scripts. For information about installing and configuring SNMP, refer to the Software Installation Guide and Configuration Guides provided with the System Release software.

SNMP related services are snmpd, boardserver, and dlctrapsrver. The dlstart script will start all the SNMP related services if SNMP is configured. The dlstop script will stop all the SNMP related services that were started by dlstart.

The ct_intel scripts in /etc/init.d support additional command line options for managing SNMP. These options are

- snmpstart – to start SNMP related services
- snmpstop – to stop SNMP related services
- snmprestart – to restart (i.e., stop and start) the SNMP related services.
- snmpstatus – to give the status of SNMP related services.

Note that dlgsnmpd is supported in this release, but will be deprecated in a future release because the options are supported by the ct_intel script. The mapping is as follows:

- dlgsnmpd start maps to ct_intel snmpstart
- dlgsnmpd stop maps to ct_intel snmpstop
- dlgsnmpd status maps to ct_intel snmpstatus
- dlgsnmpd restart maps to ct_intel snmprestart

3.3.3 Dialogic® System Software Diagnostics Guide

Update for Remote Diagnostics Package

A remote diagnostics package is now available that allows you to run Dialogic® diagnostics utilities remotely from a central site. For further information, see [Section 1.21, “Remote Diagnostics Package”](#), on page 62 of this Release Update.

Updates to **Chapter 11, DebugAngel Reference**

Because of a new feature in the Service Update, the DebugAngel tool has been enhanced to provide more capabilities for managing multiple log files. For more information about this feature, see [Section 1.7, “File Management Enhancements for DebugAngel Tool”](#), on page 39 of this Release Update.

Updates to **Chapter 20, Telecom Subsystem Summary Tool Reference**

Because of new features in the Service Update:

- The its_sysinfo tool has an option to disable collection of board memory dumps.

- The *its_sysinfo.htm* file includes a Linux Package Info section at the beginning of the file.

For further information about these features, see [Section 1.51, “Telecom Subsystem Summary Tool \(its_sysinfo\)”](#), on page 98 of this Release Update.

Update to **Chapter 24, PSTN Diagnostics Tool Reference**

An enhanced version of the PSTN Diagnostics tool (pstndiag) is provided in the Service Update. The previous version of the tool is still supported. For information about the new version, see [Section 1.22, “Enhanced Diagnostics Tools”](#), on page 64 of this Release Update.

Update to **Chapter 26, RTFManager Reference** (IPY00037518)

Section 26.5, General Tab, says that binary log format is not supported by the current release. This is not correct; binary log format is supported. For information about binary log files, see the update to **Chapter 27, Runtime Trace Facility (RTF) Reference** below.

Update to **Chapter 27, Runtime Trace Facility (RTF) Reference** (IPY00037518)

The following information about using binary log files should be added to **Section 27.3.2, Logfile Tag**:

For installations with high channel densities, or which have enabled all or most RTF trace levels, the volume of logging may result in an increased CPU utilization by the RtfServer executable as a result of the increased volume of log messages.

As shipped, the RTF log files are generated in ASCII text mode. There is a configuration parameter in the RTF configuration file (*RtfConfigWin.xml* for Windows®, *RtfConfigLinux.xml* for Linux) that allows log files to be generated in either “text” or “binary” format. Testing on high channel density systems with most or all of the RTF trace levels enabled has shown that the generation of binary format RTF log files has less of an impact on CPU usage than does the generation of text format RTF log files.

If the volume of logging results in high CPU usage, then using binary format will reduce the usage.

Enabling Binary Format RTF Log Files

The XML file contains the following line, which allows changes to log file parameters to be made:

```
<Logfile path="$(INTEL_DIALOGIC_DIR)/log" size="300" maxbackups="10" preserve_size="300"
preserve_maxbackups="10" duplicate_to_debug_console="0" log_format="text" />
```

The “log_format” value controls the type of log files that are written. Valid values for this parameter are “text” and “binary”. Once a change has been made to the XML file, it must be reloaded using the rtftool reload command.

Converting Binary Format RTF Log Files to Text Format

In order for binary log files to be examined, they must be converted into text format. This can be done by using the rtftool export command.

```
rtftool export [-d source_dir | -s source_file] [-f [dest_file] | -m dest_dir]
```

By default, the name of the text format files generated by this command will be *EXPORT-<RTF binary log file name>*. For example, if the binary format file is named *rtflog-LOCAL-20070306-15h09m26.506s.txt*, then the default name of the generated text format file will be *EXPORT-rtflog-LOCAL-20070306-15h09m26.506s.txt*. This behavior can be overridden using the *-f* command line option.

The *rtftool* utility is a stand-alone program, and it is not necessary to have the Dialogic® System Release installed on the system in order to convert RTF log files from binary to text format.

Note: When generating large binary files with RTF, do not split the single large binary file and then use the individual split files with the *rtftool* utility. *Rtftool* will not work with chopped binary files.

Update to **Chapter 28, Status Monitor Reference**

An enhanced version of the Status Monitor tool (*statusmon*) is provided in the Service Update. The previous version of the tool is still supported. For information about the new version, see [Section 1.22, “Enhanced Diagnostics Tools”](#), on page 64 of this Release Update.

3.3.4 **Dialogic® Board Management API Library Reference**

There are currently no updates to this document.

3.3.5 **Dialogic® OA&M API Programming Guide**

There are currently no updates to this document.

3.3.6 **Dialogic® OA&M API Library Reference**

There are currently no updates to this document.

3.4 **Programming Libraries Documentation Updates**

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

- [Dialogic® Audio Conferencing API Programming Guide](#)
- [Dialogic® Audio Conferencing API Library Reference](#)
- [Dialogic® Continuous Speech Processing API Programming Guide](#)
- [Dialogic® Continuous Speech Processing API Library Reference](#)
- [Dialogic® Digital Network Interface Software Reference](#)
- [Dialogic® Fax Software Reference](#)
- [Dialogic® Global Call API Programming Guide](#)
- [Dialogic® Global Call API Library Reference](#)
- [Dialogic® Global Call Analog Technology 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 Programming Guide](#)
- [Dialogic® IP Media Library API Library Reference](#)
- [Dialogic® ISDN Software Reference](#)
- [Dialogic® Learn Mode and Tone Set File API Software Reference](#)
- [Dialogic® Modular Station Interface API Programming Guide](#)
- [Dialogic® Modular Station Interface API Library Reference](#)
- [Dialogic® PBX Integration Board User's Guide](#)
- [Dialogic® PBX Integration Software Reference](#)
- [Dialogic® ADEPT for PBX Integration Boards User's Guide](#)
- [Dialogic® Standard Runtime Library API Programming Guide](#)
- [Dialogic® Standard Runtime Library API Library Reference](#)
- [Dialogic® Voice API Programming Guide](#)
- [Dialogic® Voice API Library Reference](#)

3.4.1 **Dialogic® Audio Conferencing API Programming Guide**

There are currently no updates to this document.

3.4.2 **Dialogic® Audio Conferencing API Library Reference**

There are currently no updates to this document.

3.4.3 **Dialogic® Continuous Speech Processing API Programming Guide**

Update to **Section 4.1.2, Reserving Extra Time Slots for Streaming to TDM Bus** (IPY00041018)

The paragraph about using the **CSPEExtraTimeSlot** parameter to configure the extra time slot should also include the **EC_Resource** parameter, as follows:

On Dialogic® Springware Boards in Linux, you configure this time slot at initialization time in *dialogic.cfg*. On Dialogic® Springware Boards in Windows®, you configure this time slot at initialization time in the Dialogic® Configuration Manager (DCM). Both the **CSPEExtraTimeSlot** and **EC_Resource** parameters must be enabled. See the appropriate Configuration Guide for more information about these parameters.

3.4.4 Dialogic® Continuous Speech Processing API Library Reference

Update to **ec_reciottdata()** and **ec_stream()**

The **ec_reciottdata()** and **ec_stream()** function reference pages contain a caution about channels getting stuck when failing to listen to a TDM bus time slot prior to invoking a record operation. The caution should be revised: this condition now returns an error rather than resulting in a stuck channel. The revised caution is:

- On Dialogic® DM3 Boards using a flexible routing configuration, CSP channels must be listening to a TDM bus time slot in order for the **ec_reciottdata()** and **ec_stream()** functions to work. The actual recording operation will start only after the channel is listening to the proper external time slot. In other words, you must issue a **dx_listen()** function call on the device handle before calling **ec_reciottdata()** or **ec_stream()** for that device handle, and the **dx_listen()** has to be called from the same process as the **ec_reciottdata()** or **ec_stream()**. If not, the **ec_reciottdata()** or **ec_stream()** function will return **TEC_STREAM** with **EDX_SH_MISSING** as the termination reason.

Update to **ec_reciottdata()** and **ec_stream()**

Because of a new feature in the Service Update, the **ec_reciottdata()** and **ec_stream()** functions can now be used to record and stream raw data when used with Dialogic® DM/V-A and DM/V-B Media Boards. For further information, see [Section 1.6, “Raw Data Mode Support with Dialogic® DM3 Boards”](#), on page 36 of this Release Update.

3.4.5 Dialogic® Digital Network Interface Software 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 Programming Guide

There are currently no updates to this document.

3.4.8 Dialogic® Global Call API Library Reference

Updates for analog call transfer support on Dialogic® Springware Boards

Because of a new feature in the Service Update, the **gc_BlindTransfer()**, **gc_SetupTransfer()**, **gc_CompleteTransfer()**, and **gc_SwapHold()** functions are now supported for Dialogic® Springware Analog technology. **Table 1, Global Call Function Support by Technology**, and the individual function reference pages should be updated to indicate this.

Update to **gc_GetFrame()**

In the code example, under */* Retrieve events from SRL */*
change this:

```
GC_L2_BLK recvfrmptr; /* Buffer to store received frame */
.
.
.
case GCEV_L2FRAME:
/* retrieve signaling information from queue */
if ( gc_GetFrame(ldev, &recvfrmptr) != GC_SUCCESS)
```

to this:

```
L2_BLK      l2ie;
GC_L2_BLK    recvfrmptr; /* Buffer to store received frame */
.
.
.
recvfrmptr.cclib = (void *)&l2ie;
case GCEV_L2FRAME:
/* retrieve signaling information from queue */
if ( gc_GetFrame(ldev, &recvfrmptr) != GC_SUCCESS)
```

Update to **gc_util_insert_parm_val()** (IPY00043078)

In the description for **gc_util_insert_parm_val()**, a note should be added stating that **gc_Start()** must be called before **gc_util_insert_parm_val()**. Also, the code example should be replaced with the following:

```
#include <stdio.h>
#include <srllib.h>
#include <gclib.h>
#include <gcerr.h>

void main( )
{
    GC_PARM_BLK my_blkp = NULL;
    GC_PARM_DATAP my_parmp;
    GC_INFO gc_error_info; /* GlobalCall error information data */
    int type = 1;

    /* Issue a gc_Start() call to initialize the library */
    if ( gc_Start(NULL) != GC_SUCCESS )
    {
        /* process error return as shown */
        gc_ErrorInfo( &gc_error_info );
        printf ("Error: gc_Start(), GC ErrorValue: 0x%x - %s, CCLibID: %i - %s,\n",
                GC ErrorValue: 0x%x - %s\n", gc_error_info.gcValue, gc_error_info.gcMsg,
                gc_error_info.ccLibId, gc_error_info.ccLibName,
                gc_error_info.ccValue, gc_error_info.ccMsg);
        return (gc_error_info.gcValue);
    }

    /* insert parm by reference */
    if ( gc_util_insert_parm_ref( &my_blkp, GC_SET_SERVREQ, PARM_REQTYPE,
                                sizeof( int ), &type ) != GC_SUCCESS )
    {
        /* Process error */
    }
    /* insert parm by value */
}
```

```

if ( gc_util_insert_parm_val( &my_blkp, GC_SET_SERVREQ, PARM_ACK,
                             sizeof( short ), GC_ACK ) != GC_SUCCESS )
{
    /* Process error */
}

/* Now we should have a GC_PARM_BLK with 2 parameters */

/* Following use of gc_util_next_parm retrieves the first parameter in a
 * GC_PARM_BLK, which in this case is PARM_REQTYPE */
my_parmp = gc_util_next_parm( my_blkp, NULL );

/* Retrieve the next parameter after getting the first one */
my_parmp = gc_util_next_parm( my_blkp, my_parmp );

/* This function finds and returns specified parameter, NULL if not found */
my_parmp = gc_util_find_parm( my_blkp, GC_SET_SERVREQ, PARM_ACK );

/* After GC_PARM_BLK is no longer needed, delete the block */
gc_util_delete_parm_blk( my_blkp );

/* Set my_blkp to NULL now that the block has been deleted */
my_blkp = NULL;

/* Issue gc_Stop() Next */
if (gc_Stop() != GC_SUCCESS )
{
    /* process error return as shown */
    gc_ErrorInfo( &gc_error_info );
    printf ("Error: gc_Stop(), GC ErrorValue: 0x%x - %s, CCLibID: %i - %s,
            CC ErrorValue: 0x%x - %s\n",
            gc_error_info.gcValue, gc_error_info.gcMsg,
            gc_error_info.ccLibId, gc_error_info.ccLibName,
            gc_error_info.ccValue, gc_error_info.ccMsg);
}
}

```

3.4.9 Dialogic® Global Call Analog Technology Guide

There are currently no updates to this document.

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

There are currently no updates to this document.

3.4.11 Dialogic® Global Call IP Technology Guide

Update to **Section 4.3.2, Setting Coder Information**

The description of IP_CAPABILITY.capability data structure field (page 109) and Table 2, Coders Supported for IPT Boards, (pages 110-111) both list eight value defines that are used to specify the GSM AMR-NB coder. The GSM AMR-NB coder is not supported in Dialogic® System Release 6.1 for Linux, and these values should be ignored.

Update to **Section 4.14, Sending and Receiving SIP OPTIONS Messages**

The INFO method is included as part of the Allow header in SIP messages by default. Section 4.14 should be updated in three places:

- Under **Section 4.14.1, Default OPTIONS Behavior**, change as follows:
The default Allow header will be the following if supplementary services (call transfer) is not enabled:
Allow: INVITE, CANCEL, ACK, BYE, **INFO**
or the following if supplementary services is enabled:
Allow: INVITE, CANCEL, ACK, BYE, **INFO**, REFER, NOTIFY
Note that in either case, OPTIONS is not included in the list.
- Under **Section 4.14.3, Sending OPTIONS Requests**, change as follows:
When SIP OPTIONS access is enabled, the Allow header field will be the following if supplementary services (call transfer) is not enabled:
Allow: INVITE, CANCEL, ACK, BYE, **INFO**, OPTIONS
or the following if supplementary services is enabled:
Allow: INVITE, CANCEL, ACK, BYE, **INFO**, REFER, NOTIFY, OPTIONS
The application can add additional methods to the Allow header, but the Global Call library will ensure that all of the methods supported by the library are included.
- Under **Section 4.14.6, Responding to OPTIONS Requests**, in the “**Success**” **Response Message** subsection, change as follows:
The Global Call library ensures that the Allow header field contains all SIP methods supported by the library, which includes the following methods if supplementary services (call transfer) is not enabled:
INVITE, CANCEL, ACK, BYE, **INFO**, OPTIONS
or the following if supplementary services is enabled:
INVITE, CANCEL, ACK, BYE, **INFO**, REFER, NOTIFY, OPTIONS

Update to **Section 8.2.4, IPSET_CONFIG**

New parameter IDs have been added to IPSET_CONFIG so that you can disable/enable the sending of the automatic SIP re-INVITE message upon media switch (e.g., when switching from fax to audio). For further information, see [Section 1.13, “Disabling Automatic re-INVITE Message when Switching between Fax and Audio”](#), on page 51 of this Release Update.

Updates to **Chapter 9, IP-Specific Data Structures**

In the reference section for the IP_CAPABILITY data structure (page 389), the description of the capability field lists eight value defines that are used to specify the GSM AMR-NB coder for Dialogic® IPT Boards. The GSM AMR-NB coder is not supported in Dialogic® System Release 6.1 for Linux, and these values should be ignored.

Because of a feature introduced in the Service Update, a new data structure, SIP_STACK_CFG, has been added for configuring SIP stack parameters. Related to this, the IP_VIRTBOARD data structure has been updated to point to the new structure. For further information, see [Section 1.12, “Configuring SIP Stack Parameters with Global Call”](#), on page 48 of this Release Update.

3.4.12 **Dialogic® Global Call ISDN Technology Guide**

There are currently no updates to this document.

3.4.13 Dialogic® Global Call SS7 Technology Guide

There are currently no updates to this document.

3.4.14 Dialogic® IP Media Library API Programming Guide

Dynamically Changing the Transmit Time Slot on IP Media Devices

Because of a new feature in the Service Update, a new chapter about dynamically changing the transmit time slot of IP Media devices should be added. For information about this feature, see [Section 1.27, “Dynamically Changing the Transmit Time Slot on IP Media Devices”](#), on page 71 of this Release Update.

3.4.15 Dialogic® IP Media Library API Library Reference

Update to **ipm_ModifyMedia()**

The names of the termination events given for the **ipm_ModifyMedia()** function, IPMEV_MODIFY_MEDIA and IPMEV_MODIFY_MEDIA_FAIL, are incorrect. The correct event names are IPMEV_MODIFYMEDIA and IPMEV_MODIFYMEDIA_FAIL.

Update to **ipm_SetRemoteMediaInfo()**

Because of a new feature in the Service Update, there is a new value for the **eDirection** parameter:

- DATA_MULTICAST_CLIENT – multicast client mode (supported for Dialogic® DM/IP Boards only)

For information about this feature, see [Section 1.15, “IP Multicast Client Support”](#), on page 55 of this Release Update.

Update to **ipm_StartMedia()**

Because of a new feature in the Service Update, there is a new value for the **eDirection** parameter:

- DATA_MULTICAST_CLIENT – multicast client mode (supported for Dialogic® DM/IP Boards only)

For information about this feature, see [Section 1.15, “IP Multicast Client Support”](#), on page 55 of this Release Update.

Update to **Chapter 3, Events**

The names of the termination events given for the **ipm_ModifyMedia()** function, IPMEV_MODIFY_MEDIA and IPMEV_MODIFY_MEDIA_FAIL, are incorrect. The correct event names are IPMEV_MODIFYMEDIA and IPMEV_MODIFYMEDIA_FAIL.

Update to IPM_PARM_INFO data structure

Because of a new feature in the Service Update, a new parameter called PARMCH_TX_TIMESLOT should be added in IPM_PARM_INFO data structure. For information about this feature, see [Section 1.27, “Dynamically Changing the Transmit Time Slot on IP Media Devices”](#), on page 71 of this Release Update.

3.4.16 Dialogic® ISDN Software Reference

Update to **cc_GetDLinkState()**

In the reference information for the **cc_GetDLinkState()** function, the description paragraph should read: "The **cc_GetDLinkState()** function retrieves the logical data link state (operable or inoperable) of the specified board device for PRI or station device for BRI."

Update to **cc_GetDLinkState()** (PTR# 25745)

In the description of the **state_buf** parameter for the **cc_GetDLinkState()** function, only two possible data link states are defined: DATA_LINK_UP and DATA_LINK_DOWN. DATA_LINK_DISABLED is not a valid value.

Update to **cc_GetEvtMsk()**

In the **cc_GetEvtMsk()** function reference pages, **Table 20, Bitmask Values**, incorrectly indicates the default values for CCMSK_SERVICE_ACK and CCMSK_SETUP_ACK as "Not enabled". The correct default values are "Enabled."

Update to **cc_GetEvtMsk()** (PTR# 29203)

In the **cc_GetEvtMsk()** function reference pages, **Table 20, Bitmask Values**, incorrectly lists CCMSK_TERMINATE as a supported bitmask type. The CCMSK_TERMINATE bitmask type is not supported.

Update to **cc_GetEvtMsk()** (PTR# 29036)

The descriptions of the CCMSK_TMREXPEVENT bitmask in the **cc_GetEvtMsk()** and **cc_SetEvtMsk()** functions mention that the CCEV_TIMER event is generated when a Layer 3 timer expires, but there is no description of how to retrieve the Timer ID and Call ID values associated with the CCEV_TIMER event. The following text describes how to retrieve these values with the assumption that the CCEV_TIMER event has been enabled:

In the application, define a TIMER_DATA structure as follows:

```
typedef struct _TIMER_DATA {
    unsigned char tbd_1; // 0
    unsigned long CallId; // 1 2 3 4
    unsigned short TimerId; // 5 6
    unsigned short tbd_2; // 7 8
}TIMER_DATA, *PTIMER_DATA;
```

Then, retrieve the values as follows:

```
(evtdatap = sr_getevtdatap(...))
case CCEV_TIMER:
{
    PTIMER_DATA pData = (PTIMER_DATA)evtdatap;
    m_TimerCallId = pData->CallId;
    m_TimerId = pData->TimerId;
    Log(MSG_EVENT,"Timer: Call_id = %d, Timer expired ID = (%d) 0x%x",
        m_TimerCallId, m_TimerId);
}
break;
.
.
.
```

Update to **cc_GetParmEx()** (PTR# 35969)

In the **cc_GetParmEx()** function reference page, the last entry in **Table 22, cc_GetParmEx() Parameter ID Definitions** (page 145) incorrectly lists SUBADDR_NUMBER as a parameter definition. There is no such parameter definition; the correct parameter definition is SUBADDRESS_NUMBER.

Update to **cc_MakeCall()** (PTR# 22842)

In the **cc_MakeCall()** function reference information, the description of the **numberstr** parameter should read: The destination (called party's) telephone number string. The maximum number of digits is dictated by the protocol switch specification. Users need to find out the specification limits for the protocol they wish to use, otherwise the protocol stack will reject the request to make a call.

Update to **cc_MakeCall()** and **cc_SetCallingNumber()** (PTR# 28720)

The following caution should be included in the **cc_MakeCall()** and **cc_SetCallingNumber()** function reference pages:

- When using **cc_MakeCall()** to make an outbound call, if the **origination_phone_number** field in the **MAKECALL_BLK** structure is set to NULL or '\0' (null string), the **destination_number_plan** and the **destination_number_type** fields in the **MAKECALL_BLK** structure are ignored. This precludes the option of using the **cc_SetCallingNumber()** function to set the origination phone number and specifying a value of NULL or '\0' for the **origination_phone_number** field in the **MAKECALL_BLK** structure, when the **destination_number_plan** and the **destination_number_type** values (as specified in the **destination_number_plan** and **destination_number_type** fields in the **MAKECALL_BLK** structure) must be included in the outgoing message.

Update to **cc_SetEvtMsk()**

In the **cc_SetEvtMsk()** function reference pages, **Table 24, Bitmask Values**, incorrectly indicates the default values for **CCMSK_SERVICE_ACK** and **CCMSK_SETUP_ACK** as "Not enabled". The correct default values are "Enabled."

Update to **cc_SetEvtMsk()** (PTR# 29036)

The descriptions of the **CCMSK_TMREXPEVENT** bitmask in the **cc_GetEvtMsk()** and **cc_SetEvtMsk()** functions mention that the **CCEV_TIMER** event is generated when a Layer 3 timer expires, but there is no description of how to retrieve the Timer ID and Call ID values associated with the **CCEV_TIMER** event. The following text describes how to retrieve these values with the assumption that the **CCEV_TIMER** event has been enabled:

In the application, define a **TIMER_DATA** structure as follows:

```
typedef struct _TIMER_DATA {
    unsigned char tbd_1; // 0
    unsigned long CallId; // 1 2 3 4
    unsigned short TimerId; // 5 6
    unsigned short tbd_2; // 7 8
}TIMER_DATA, *PTIMER_DATA;
```

Then, retrieve the values as follows:

```
(evtdatap = sr_getevtdatap(...))
case CCEV_TIMER:
{
    PTIMER_DATA pData = (PTIMER_DATA)evtdatap;
    m_TimerCallId = pData->CallId;
```

```

        m_TimerId = pData->TimerId;
        Log(MSG_EVENT, "Timer: Call_id = %d, Timer expired ID = (%d) 0x%x",
            m_TimerCallId, m_TimerId);
    }
    break;
    .
    .
    .

```

Update to **cc_SetParmEx()**

In the **cc_SetParmEx()** function reference page, beginning on page 250, the parm_id description in the parameters table (page 251) incorrectly identifies SUBADDR_NUMBER as a supported parameter. There is no such parameter; the correct parameter name is SUBADDRESS_NUMBER.

3.4.17 **Dialogic® Learn Mode and Tone Set File API Software Reference**

There are currently no updates to this document.

3.4.18 **Dialogic® Modular Station Interface API Programming Guide**

There are currently no updates to this document.

3.4.19 **Dialogic® Modular Station Interface API Library Reference**

There are currently no updates to this document.

3.4.20 **Dialogic® PBX Integration Board User's Guide**

There are currently no updates to this document.

3.4.21 **Dialogic® PBX Integration Software Reference**

There are currently no updates to this document.

3.4.22 **Dialogic® ADEPT for PBX Integration Boards User's Guide**

Update to terminology in the User's Guide

After the bullet list on page 8 in Section 2.1, all subsequent references to "*libd42mt.dll*" and "*libd42mt.lib*" should be replaced with "*D42 library*".

The D42 library is used as a collective term for Windows® and Linux:

- For Windows® – *libd42mt.dll* is the runtime library file (needed to run an application) and *libd42mt.lib* is the compile time library file (needed to compile the application).

- For Linux – runtime and compile time libraries are the same and the file name is *libd42.so*.

Update to **Section 1.2, Files Changed/Added**

The following additions should be made:

The files that were changed or added to the Dialogic® System Release for ADEPT Linux implementation are listed below:

File name	Added/Changed	Location
Hicom.adt	Added	/usr/dialogic/cfg
Lucent.adt	Added	/usr/dialogic/cfg
Mitel.adt	Added	/usr/dialogic/cfg
Nec.adt	Added	/usr/dialogic/cfg
Ntbcm.adt	Added	/usr/dialogic/cfg
Ntm1.adt	Added	/usr/dialogic/cfg
Rolm.adt	Added	/usr/dialogic/cfg
libd42.so	Changed	/usr/dialogic/lib
/usr/lib		
d42lib.h	Changed	/usr/dialogic/inc
d82u.fwl	Changed	/usr/dialogic/data
d82ucsp.fwl	Changed	/usr/dialogic/data
d42ucsp.fwl	Changed	/usr/dialogic/data
Lucent_2_wire.fwl	Changed	/usr/dialogic/data
Lucent_4_wire.fwl	Changed	/usr/dialogic/data
Mitel_DNIC_M430.fwl	Changed	/usr/dialogic/data
Mitel_DNIC_M420.fwl	Changed	/usr/dialogic/data
NEC_DTerm_III.fwl	Changed	/usr/dialogic/data
Nortel_BCM.fwl	Changed	/usr/dialogic/data
Nortel_Meridian_1.fwl	Changed	/usr/dialogic/data
Nortel_Norstar.fwl	Changed	/usr/dialogic/data
Siemens_Hicom.fwl	Changed	/usr/dialogic/data
Siemens_Rolm.fwl	Changed	/usr/dialogic/data
Siemens_rolm_9006.fwl	Changed	/usr/dialogic/data

Update to **Section 2.1, The ADEPT module in D42 library**

Replace the first two paragraphs and bullet list with the following:

The ADEPT functionality is implemented in the D42 library (*libd42mt.dll* and *libd42mt.lib* files for Windows® and *libd42.so* for Linux) with the appropriate constant declared in the *d42lib.h* file.

A rebuild of the application with the new *libd42mt.lib/libd42.so* and *d42lib.h* is necessary to enable the ADEPT functionality.

- *libd42mt.dll* resides in:
windows\system32 folder and <installation folder>\dialogic\lib folder

- *libd42mt.lib* resides in:
<installation folder>\dialogic\inc
 - *libd42.so* resides in:
/usr/dialogic/lib and /usr/lib
 - *d42lib.h* resides in:
<installation folder>\dialogic\inc (for Windows®) and /usr/dialogic/inc (for Linux)
- See [Section 1.26, “New ANI/DNIS-Enabled Parsing Tool \(ADEPT\) for Dialogic® PBX Integration Boards”](#), on page 70 of this Release Update for more information.

3.4.23 Dialogic® Standard Runtime Library API Programming Guide

There are currently no updates to this document.

3.4.24 Dialogic® Standard Runtime Library API Library Reference

Update to **sr_getfdcnt()** and **sr_getfdinfo()** (IPY00045054)

The following caution should be added for **sr_getfdcnt()** and **sr_getfdinfo()**:

- The application must call **sr_getfdcnt()** or **sr_getfdinfo()** before calling any other Dialogic® API, if the application wants to use SELECT to retrieve SRL events on Linux.

3.4.25 Dialogic® Voice API Programming Guide

Functions not supported

The **r2_creatfsig()** and **r2_playbsig()** functions, which were previously provided for backward compatibility only, are no longer supported. All references to these functions should be deleted. R2MF signaling is typically accomplished through the Dialogic® Global Call API.

Update to **Chapter 6, Application Development Guidelines**

The following note should be added to **Section 6.4.2, Multithreading and Multiprocessing**:

Note: The continuous speech processing architecture allows a voice channel to be shared between processes (or applications) on Dialogic® JCT Boards, on Dialogic® DM3 Boards, or on Dialogic® Host Media Processing (HMP) (starting with Dialogic® Host Media Processing Software Release 1.3 for Windows®), providing one process does the play activity and the other process does the record/stream activity. Other CSP scenarios are **not** supported, such as playing or recording/streaming from both processes. For details, refer to the application note, Telephony Application Architectures for Dialogic® Boards with DM3 Architecture, located at http://www.dialogic.com/products/tdm_boards/media_processing/docs/9380an.pdf.

Update to **Section 13.1.10, Guidelines for Creating User-Defined Tones** (PTR# 35667)

The following note should be added:

Note: In the case where the number of tone templates is exhausted, no error is returned on **dx_addtone()** and subsequent tone detection may fail.

Update to **Section 13.1.10, Guidelines for Creating User-Defined Tones** (PTR# 34546)
The following guideline should be added:

- On Dialogic® DM3 Boards, building and adding tones of zero frequency values to a tone template can cause firmware failures.

Update to **Section 14.3, Enabling Global DPD**

Because of a new feature in the Service Update, it is no longer necessary to order a separate GDPD enablement package to enable Global Dial Pulse Detection on a board. Information about the GDPD enablement package should be removed from this section. See [Section 1.16, “Global DPD Enabled on Dialogic® Springware Boards”](#), on page 57 of this Release Update for further information.

3.4.26 Dialogic® Voice API Library Reference

Functions not supported

The **r2_creatfsig()** and **r2_playbsig()** functions, which were previously provided for backward compatibility only, are no longer supported. All references to these functions should be deleted. R2MF signaling is typically accomplished through the Dialogic® Global Call API.

Update to **dx_deltone()** function (IPY00079097)

The following caution should be added for **dx_deltone()**:

- With Dialogic® Springware Boards, calling **dx_deltone()** suspends all tone detection on the specified channel, including DTMF tone detection; it resumes upon **dx_deltone()** return. The user may experience missed or duplicated DTMF digits if **dx_deltone()** is called while a DTMF digit is being detected. It is recommended that you call **dx_clrdigbuf()** after **dx_deltone()** to clear any inappropriately detected DTMF digits from internal digit buffer.

Update to **dx_getdig()** (IPY00038453)

For Dialogic® DM3 Boards, the return value of **dx_getdig()** in synchronous mode has been changed to return 0 instead of 1 when there are no digits in the buffer. The NULL character in the digit string 'dg_value' is no longer counted as a digit. Similarly, when **dx_getdig()** returns the number of digits, the terminating NULL is no longer added to the number of digits. (The NULL was previously counted in the numdig return value calculation, but since it is not a digit, the NULL is no longer included.)

For Dialogic® Springware Boards, the terminating NULL **is** included in the number of digits. So for Springware Boards, **dx_getdig()** still returns 1 when there are no digits in the buffer.

Update to **dx_OpenStreamBuffer()** (IPY00044981)

The following caution should be added for **dx_OpenStreamBuffer()**:

- When using Dialogic® DM3 Boards, the **dx_open()** function must be called on a board, channel, or physical board before **dx_OpenStreamBuffer()** is called. Failure to do so would prevent the DM3 library from loading, and **dx_OpenStreamBuffer()** would fail.

Update to **dx_rec()**, **dx_reciottdata()**, **dx_recvox()**, and **dx_recwav()**

The **dx_rec()**, **dx_reciottdata()**, **dx_recvox()**, and **dx_recwav()** function reference pages contain a caution about channels getting stuck when failing to listen to a TDM bus time slot prior to invoking a record operation. The caution should be revised: this

condition now returns an error rather than resulting in a stuck channel. The revised caution is:

- On Dialogic® DM3 Boards using a flexible routing configuration, voice channels must be listening to a TDM bus time slot in order for voice recording functions, such as **dx_reciottdata()** and others, to work. The actual recording operation will start only after the voice channel is listening to the proper external time slot. In other words, you must issue a **dx_listen()** function call on the device handle before calling a voice recording function for that device handle, and the **dx_listen()** has to be called from the same process as the voice recording function. If not, the voice recording function will return TDX_ERROR with EDX_SH_MISSING as the termination reason.

Update to **dx_setchxfercnt()**

Because of a feature introduced in the Service Update, the **dx_setchxfercnt()** function has two new values for the **bufsize_identifier** parameter to set the buffer size to 256 or 512 bytes. For information about these new parameter values, see [Section 1.17, "Setting Data Transfer Buffer Size below 1K for Dialogic® Springware Boards"](#), on page 57 of this Release Update.

Update to **dx_setevtmask()** (IPY00038053)

The following information should be added to the description of the **mask** parameter:

User defined tones that are associated an optional digit (**dx_addtone()**) have digit reporting enabled by default in Dialogic® System Release 6.1 for Linux. The user defined tones digit reporting can be turned off by using **dx_setevtmask()** with DM_DIGOFF mask. To reactivate digit reporting, use **dx_setevtmask()** with DM_DIGITS mask.

Update to **dx_setparm()**

In **Table 16, Voice Channel Parameters (Springware)**, the description of the **DXCH_XFERBUFSIZE** parameter should be revised as follows:

Sets the size of both the play and record buffers used to transfer data between the application on the host and the driver. This parameter can be used with the **dx_getparm()** function. The largest available buffer size is 32 Kbytes (must be in multiples of 128); however, only certain discrete buffer size values are supported. Please refer to the **dx_setchxfercnt()** function where the actual buffer sizes are documented, including the smallest size allowed.

Update to **dx_setsvcond()** function (IPY00079103)

The description for the **dx_setsvcond()** function should include the following note:

Note: The pause and resume play feature is not supported on Dialogic® Springware Boards.

Update to DX_SVCB data structure (IPY00079103)

The description for the DX_SVCB data structure should include the following note:

Note: The pause and resume play feature is not supported on Dialogic® Springware Boards.

In addition, in the field description for the type field, the descriptions of the SV_PAUSE and SV_RESUME options should be changed as follows:

- SV_PAUSE - For Dialogic® DM3 Boards only; use with SV_SPEEDTBL to pause the play on detection of the specified DTMF digit.

- **SV_RESUME** - For Dialogic® DM3 Boards only; use with **SV_SPEEDTBL** to resume the play on detection of the specified DTMF digit.

Update to DX_XPB data structure

Because of a new feature in the Service Update, the DX_XPB data structure has a new value for the wDataFormat field, **DATA_FORMAT_RAW**. This value is applicable to Dialogic® DM/V-A and DM/V-B Media Boards. For further information, see [Section 1.6, “Raw Data Mode Support with Dialogic® DM3 Boards”](#), on page 36 of this Release Update.

3.5 Demonstration Software Documentation Updates

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

- [Dialogic® Continuous Speech Processing API Demo Guide](#)
- [Dialogic® CT Bus Clock Fallback Demo Guide](#)
- [Dialogic® Global Call API Demo Guide](#)
- [Dialogic® High Availability for Linux Demo Guide](#)
- [Dialogic® IP Media Server \(Global Call\) Demo Guide](#)
- [Dialogic® Voice API for Linux Demo Guide](#)

3.5.1 Dialogic® Continuous Speech Processing API Demo Guide

There are currently no updates to this document.

3.5.2 Dialogic® CT Bus Clock Fallback Demo Guide

There are currently no updates to this document.

3.5.3 Dialogic® Global Call API Demo Guide

There are currently no updates to this document.

3.5.4 Dialogic® High Availability for Linux Demo Guide

There are currently no updates to this document.

3.5.5 Dialogic® IP Media Server (Global Call) Demo Guide

There are currently no updates to this document.

3.5.6 Dialogic® Voice API for Linux Demo Guide

There are currently no updates to this document.

3.6 Pigeon Point Systems Linux Hot Swap Kit Documentation Updates

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

- [Pigeon Point Systems Linux Hot Swap Kit User Guide](#)

3.6.1 Pigeon Point Systems Linux Hot Swap Kit User Guide

The following is supplemental information for this document, which is a PDF file located in the *redistributable-runtime/PPS* directory.

Performing Switchover

Follow these steps to perform switchover:

1. Select “1 – RH Commands”
2. Select “3 – Switchover Commands”
3. Choose one of the following
 - 1 Execute Fully Cooperative Switchover” – Active host – shutdown all Software (drivers) before giving up the domain
 - 2 Execute Partially Cooperative Switchover” - Active host - shutdown some of Software (drivers) before giving up the domain
 - 3 Execute Forced Switchover” – Active host take the PCI bus without informing Software (drivers) before giving up the domain

4. The following should be seen:

```
Domain 0 belongs to host 9
Domain 1 belongs to host 9
Domain 0 is disconnected
Domain 1 is disconnected
Preparing for switchover to host 11
Domain 0 is disconnected
Domain 1 is disconnected
Switchover successful
```

(This assumes 9 was ACTIVE and 11 was STANDBY.)

If there is a failure, check the state of the domains (refer to “Checking the State of Domains.”)

Checking the State of Domains

Follow these steps to check domain ownership:

1. Select “1 – RH Commands”
2. Select “1 – Domain Information”
3. Select “3 – Get Domain Ownership”
4. You should see the following:

```
Domain #0 is owned by host #9"
Domain #1 is owned by host #9"
```

****or**:**

```
Domain #0 is owned by host #11"
Domain #1 is owned by host #11"
```

This tells you which side is ACTIVE and which is STANDBY.

If you see the following, it means that there is a Hot Swap Kit/Redundant Host configuration problem

```
Domain #0 is owned by host #9"
Domain #1 is owned by host #11"
```

(or vice versa)

If you see the above, the Dialogic® System Release software will fail on boards or a panic may occur. In this case, a cool boot is required. Switchover will make the problem worse even if the domains are corrected.