Intel NetStructure® Host Media Processing Software Release 1.2 for Linux

Release Update

September 2006

05-2352-005

Intel may make changes to specifications and product descriptions at any time, without notice.

This Intel NetStructure® Host Media Processing Software Release 1.2 for Linux Release Update as well as the software described in it is furnished under license and may only be used or copied in accordance with the terms of the license. The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document.

Publication Date: September 2006

Document Number: 05-2352-005

Intel
1515 Route 10
Parsippany, NJ 07054

For **Technical Support**, visit the Intel Telecom Support Resources website at:
http://developer.intel.com/design/telecom/support

For **Products and Services Information**, visit the Intel Telecom and Compute Products website at:
http://www.intel.com/design/network/products/telecom

For **Sales Offices** and other contact information, visit the Buy Telecom Products page at:
http://www.intel.com/buy/networking/telecom.htm

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 Intel® NetStructure™ Host Media Processing Software Release 1.2 for Linux. In addition to summarizing issues that were known as of this release, the Release Update will continue to be updated to serve as the primary mechanism for communicating new issues that arise after the release date.

Intended Audience

This Release Update is intended for all users of Intel® NetStructure™ Host Media Processing Software Release 1.2 for 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 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 performance of the HMP software. The lists include both known issues as well as issues that have been resolved since the last release. Also included are restrictions and limitations that apply to this release, as well as notes on compatibility.
- Documentation Updates: This section contains corrections and other changes that apply to the HMP software release documentation set that could not be 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 further information on issues that have an associated defect number, you may use the web-enabled Defect Query tool at:

  http://membersresource.intel.com/defects/

  When you select this link, you will be asked to either LOGIN or JOIN. If you have any questions about this tool or wish to provide feedback regarding this tool, please use the contact page on the Telecom Support Services Web site at:

  http://resource.intel.com/telecom/support/webcontact.htm

- If you are new to the Intel® telecom products or the latest technology and do not want to search through volumes of documentation looking for information, use the Learn About tool to get started quickly. This tool is located:

  http://resource.intel.com/telecom/support/documentation/learnabout/index.htm

- For technical support resources:

  http://developer.intel.com/design/telecom/support/

- For general information about Intel telecom products:
  http://www.intel.com/design/network/products/telecom/index.htm

  For information about the features supported in this release, see the *Release Guide* that is included as part of the documentation bookshelf for this release.

Document Revision History

This revision history summarizes the changes made in each published version of the Release Update for Intel® NetStructure™ Host Media Processing Software Release 1.2 for Linux, which is a document that is subject to updates during the lifetime of the release.

Document Rev 05, published September 2006

Updated for **Service Update 18**.

Note: This Revision History and the Release Issues sections have been modified to show issues by Change Control System **defect number** and by PTR number. Issues reported prior to March 27, 2006, will be identified by both numbers. Issues reported after March 27, 2006, will only have a defect number.

The Release Issues section contains the following changes:

- Added the following items to Table 1, "Issues Resolved in HMP 1.2", on page 15: IPY00031503 (PTR 36781), IPY00032235 (PTR 36405), IPY00032236 (PTR 36062), IPY00033758, IPY00033948.

The Documentation Updates section contains the following changes:

- In the Development Software Documentation section, updated the Audio Conferencing API for Linux and Windows Operating Systems Library Reference and the Audio Conferencing API for Linux Operating Systems Programming Guide to indicate that the MSG_ACTID parameter enables/disables active talker identification (or notification) and not the active talker feature itself. (IPY00006584 = PTR 36199).
- In the Development Software Documentation section, updated the Global Call IP for Host Media Processing Technology Guide to correct information on the gc_MakeCall( ) Variances for IP. (IPY00029956 = PTR 36646). Also revised the descriptions for the IP_H221NONSTANDARD data structure fields.

Document Rev 04, published December 2005

Updated for **Service Update 14**.

The Release Issues section contains the following changes:

- Added the following items to Table 1, "Issues Resolved in HMP 1.2", on page 15: IPY00032236 (PTR 36062), IPY00032235 (PTR 36405), IPY00023802 (PTR 36484).

Document Rev 03, published June 2005

Updated for **Service Update 9**.

The Post-Release Developments section includes the following changes:

- Added support for Transaction Record. See Section 1.6, “Transaction Record”, on page 12.

The Release Issues section contains the following changes:

- Added the following items to Table 1, “Issues Resolved in HMP 1.2”, on page 15: IPY00030253 (PTR 34363), IPY00009813 (PTR 34638), IPY00010596 (PTR 34758), IPY00009054 (PTR 34875).

The Documentation Updates section contains the following changes:

- In the Release Documentation section, for the Release Guide, corrected the Active Talker information in the Features list and removed **dcb_monconf()** and **dcb_unmonconf()** from the list of Audio Conferencing API functions not supported.
- In the Operating Software Documentation section, added information about starting the License Manager to the Intel NetStructure® Host Media Processing Software Release 1.2 for Linux License Manager Administration Guide.
- In the Development Software Documentation section, added information about the **dx_mreciottdata( )** function to the Voice API Host Media Processing Library Reference.
- In the Development Software Documentation section, added information about Transaction Record to the Voice API for Host Media Processing Programming Guide.
- In the Demonstration Software Documentation section, for Linux, the path names and executables for starting all of the demos have been changed.

Document Rev 02, published January 2005

Updated for **Service Update 6**.

Added a new section, Post-Release Developments, that describes the new features contained in the Service Update and provides information about installing the Service Update software.

The Release Issues section contains the following changes:

- Added the following items to Table 1, “Issues Resolved in HMP 1.2”, on page 15: IPY00009304 (PTR 33638), IPY00009801 (PTR 33644), IPY00009082 (PTR 33762), IPY00009204 (PTR 33894), IPY00030251 (PTR 33919), IPY00010238 (PTR 34148).
- Removed a number of Global Call, IP, and IPML Restrictions and Limitations which have either been resolved or documented elsewhere.
- Added new Operating System Notes section

The Documentation Updates section contains the following changes:

- In the Release Documentation section, added information about revising the footnote in Table 1 of the Release Guide.
- In the Release Documentation section, provided new Software Requirements.

- In the Release Documentation section, added information about revising Table 2 of the Release Guide to support 120 low bit rate coders (Enhanced RTP resources).
- In the Demonstration Software Documentation section, added note with respect to each Demo Guide about demos not working on channel numbers above 120.

Document Rev 01, published October 2004

Initial version of document.

Post-Release Developments 1

This section describes significant changes to the Intel® NetStructure™ Host Media Processing Software Release 1.2 for Linux subsequent to the general availability release date.

1.1 Service Update for Host Media Processing Software 1.2 for Linux

A Service Update for Intel® NetStructure™ Host Media Processing Software Release 1.2 for Linux is now available. Service Updates provide fixes to known problems, and may also introduce new functionality. New versions of the Service Update will be released periodically. This Release Update will document the features in the Service Update.

Depending on whether you already have an Intel® NetStructure™ Host Media Processing Software Release 1.2 for Linux Service Update installed on your system, the Service Update will either give you a **full install** or an **update install**.

If you do not have Intel® NetStructure™ Host Media Processing Software Release 1.2 for Linux installed on your system, installing the Service Update will provide a **full install** of Intel® NetStructure™ Host Media Processing Software Release 1.2 for Linux.

If you already have an Intel® NetStructure™ Host Media Processing Software Release 1.2 for Linux Service Update installed on your system, installing this Service Update will provide an **update install**. The update install adds the latest software to support the features provided by the Service Update.

1.2 Service Update Software Installation

The following information supercedes that of the Intel® NetStructure™ Host Media Processing Software Release 1.2 for Linux Installation Guide.

1.2.1 Assumptions and Prerequisites

The following are basic requirements to install and run the HMP software:

- Before installing the HMP Software, make sure that the system meets the hardware and software “System Requirements” as described the HMP Software *Release Guide*. These requirements include the following:
  - A system with **either** Red Hat Enterprise Linux including Update 1 or Update 3 installed (You must also install the Intel-provided 2.6.8.1 kernel on top of the base Red Hat release. See the README file of this HMP release.) **or** SUSE Linux Enterprise Server installed (includes kernel version 2.6.8-24).
  - A minimum of 512MB of RAM if running without an X Windows System.
  - A minimum of 1GB of RAM if running in an X Windows System.
  - The machine must be able to resolve its name to an IP address, either using DNS or the /etc/hosts file. If it cannot resolve its name, HMP will NOT start.
- To install the software, you must have local administrative privileges. Contact your network administrator to set up administrative privileges as required.
- Uninstall any previous version of the HMP Software (or any Intel® Dialogic® system release).

Note that the HMP Service Update software has three install scripts:

- One at the top level which installs both runtime components as well as header files (run this one to be able to compile your application) (recommended)
- One in the redistributable-runtime directory, which installs runtime components (no header files)
- One in the sdk directory, which installs header files for application development

1.2.2 Order of Procedures

This section describes the order in which the installation procedures must be performed.

If this is a new install, before you use the HMP Software, you must obtain a license file containing HMP license data. You can obtain a license before or after you install the HMP Software. A verification license supplied with the software will allow you to run the verification demo (IP Media Server demo) to confirm that you have installed the HMP Software properly. However, its features are limited and you may want to obtain another license. For more information, refer to the *Host Media Processing Software Release 1.2 for Linux License Manager Administration Guide*.

The basic order of the installation process is as follows:

1. Install the HMP 1.2 Service Update software as described in Section 1.2.3, “Installing the Service Update”, on page 10.
2. Use the License Manager application to configure the system. This consists of activating an HMP Software license (refer to the *Host Media Processing Software 1.2 for Linux Installation Guide*).

3. Manually start HMP services using dlstart. HMP services will start automatically whenever you reboot your system.

1.2.3 Installing the Service Update

The following procedure describes how to install the Service Update.

1. Log in to the Linux system as root.

2. Download and unarchive the Service Update software from the Web.

3. Enter the following command to start the install script:

   ```
   ./install.sh
   ```

   The following messages are displayed:

   ```
   -----------------------------------------------------------------------
   Installing Intel(R) NetStructure(TM) Host Media Processing Software 1.2
   Redistributable Runtime for Linux

   Initializing install, please wait .......

   =======================================================================
           Intel ® NetStructure ™ Host Media Processing Software 1.2
                     Redistributable Runtime
                          INSTALLATION

   You will now have the opportunity to install software packages.
   After the menu is displayed, enter the package number(s) of the desired
   packages, separated by a space. Enter A for all packages, Q to quit.

   Package dependencies will be automatically resolved during installation. For
   example, selecting a single package will automatically install all packages
   required for that selection.

   Press ENTER to display the menu packages.
   ```

4. Press **Enter**.

   The following menu is displayed:

   ```
   Item   Package Description
   ---- ------------------------------------------------------------------
     1    Intel(R) NetStructure(TM) Host Media Processing Software (74 MB)
     2    SNMP Component Manager (48 MB)
     3    Documentation (15 MB)

     A    Install All (96 MB)
     Q    Quit Installation

   Enter the packages you wish installed, separated by a space, or [A,a,Q,q]:
   ```

5. Enter the numbers or letter that correspond to what you want to install. Unless space or other considerations dictate that you limit what you install, you can just enter **a** to install everything.

   You will see messages about package installation order, checking for previously installed packages, checking for sufficient disk space, and installation progress. Then the installation menu (shown above) will reappear.

6. Enter **q** to quit and complete the installation process. The following message is displayed:

   ```
   Do you wish to run config.sh to configure your system [Yn] ?
   ```

7. At this point, you have two options:
   - Enter ***y*** if you already have obtained a license or want to use the verification license supplied with the software. Follow the procedure for activating a license in Section 2.4 of the *Host Media Processing Software Release 1.2 for Linux Installation Guide*.
   - Enter ***n*** if you do not want to use the verification license or have not obtained the license you want. After you have obtained a license, run *config.sh* (located in */usr/dialogic/bin*) and follow the procedure for activating a license in Section 2.4 of the *Host Media Processing Software Release 1.2 for Linux Installation Guide*.

   Note: You must run `config.sh` at least once after installing the software.

After the configuration has been completed, the following messages are displayed:

```
Configuration is now complete.

Before using the software, you must ensure that the Intel(R) Dialogic(R)
environment variables are set using the following action:

       - Logout and login

The Intel(R) NetStructure(TM) system services will automatically
start every time the system is reboooted.

NOTE: To start and stop system services manually, use the dlstop and
dlstart scripts found in /usr/dialogic/bin.


Do you wish to run the HMP demo on your system [y/n] ?
```

Enter **y** to run the HMP Verification Demo. The system will then start the Intel Dialogic services. After the Intel Dialogic services have been started, the HMP Verification Demo will start automatically. After the demo has completed, the Intel Dialogic services will be stopped and the following message should be displayed:

```
Installation of the Intel(R) NetStructure(TM) Host Media Processing Software 1.2
Redistributable Runtime for Linux was successful.
```

1.2.4 HMP Verification Demo

The HMP Verification Demo confirms the success of the HMP Software Service Update 1 installation by establishing a basic level of functionality after the installation has been completed. The demo performs some sanity checks and then waits for a Voice Over Internet Protocol (VoIP) call to be placed using an H.323 or SIP phone configured to use G.711. After the connection is made, a message prompt is played to the calling party.

Note: The HMP Verification Demo can only be launched from the /usr/dialogic/demos/verification_demo location. The demo will fail if launched from a different directory.

Note: If you are using Microsoft* NetMeeting* with Windows* XP* to place a call, there may be up to one minute delay from the time a call is placed until the prompt is played.

1.2.5 Environment Variables

The INTEL_DIALOGIC_LIB environment variable (for example, -L${INTEL_DIALOGIC_LIB} -lgc) should be used for linking to Intel HMP 1.2 Service

Update 1 libraries. Failure to use this environment variable in *makefiles* may cause linking failures due to potential name conflicts.

1.3 Support for SUSE Linux Enterprise Server 9.2

The Service Update also provides support for SUSE Linux Enterprise Server 9.2.

At a minimum, the SUSE Linux Enterprise Server kernel *.config* file should be configured as follows:

- CONFIG_M586 should be disabled and CONFIG_MPENTIUM4 or CONFIG_MPENTIUMIII should be enabled, depending on your processor
- CONFIG_PREEMPT should be enabled
- CONFIG_HPET_RTC_IRQ should be disabled

Although the *.config* file can be manually configured, it is recommended that you use the provided configuration script from the Web site:

http://resource.intel.com/telecom/support/hmp11/index.htm

1.4 Support for Red Hat Enterprise Linux Update 3

In addition to Red Hat Enterprise Linux Update 1, Update 3 is now also supported by the Service Update.

Note: Update 2 is **not** supported by this Service Update.

1.5 Support for 120 Low Bit Rate Channels

This Service Update increases the number of low bit rate channels supported to 120. See the revised Table 2 in Section 3.2.1, “Host Media Processing Software Release 1.2 for Linux Installation Guide”, on page 28.

1.6 Transaction Record

Transaction Record enables the recording of a two-party conversation by allowing two time slots from a single channel to be recorded. This feature is useful in call center applications where it is necessary to archive a verbal transaction or record a live conversation. Voice activity on two channels can be summed and stored in a single file or combination of files, devices, and memory.

The voice API function **dx_mreciottdata()** is used to implement the transaction record feature. It is an extension of the **dx_reciottdata()** function.

For additional Information about Transaction Record, see *Section 3.4.14, "Voice API for Host Media Processing Library Reference"*, on page 34 and *Section 3.4.15, "Voice API for Host Media Processing Programming Guide"*, on page 41.

Release Issues 2

This chapter includes the following topics that relate to release issues:

2.1 Issues

The following table lists issues that can affect the performance of the software supported in Intel® NetStructure™ Host Media Processing Software Release 1.2 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** - An issue that affects the performance of the Host Media Processing (HMP) Software. This category includes interoperability issues and compatibility issues. Known issues are still open but may or may not be fixed in the future. See Table 2.
- **Resolved** - An issue that was resolved in a previous release of the product (usually either fixed or documented). See Table 1.

Defect Number

A unique identification number that is used to track each issue reported via a formal Change Control System. Additional information on defects may be available via the Defect Query tool at *http://membersresource.intel.com/defects/* (If you select this link, you will be asked to login. If you do not have a member login, you may create one.) For issues without an associated defect number, this column contains “NA” (Not Applicable).

PTR Number

Problem Tracking Report 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. (The Defect Query tool also provides the ability to search by legacy PTR number.)

Product or Component

The product or component to which the problem relates, typically one of the following:

- a system-level component; for example, HMP 1.2 Linux
- a software product; for example, the Global Call Library
- a software component; for example, Continuous Speech Processing

Description

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

Table 1. Issues Resolved in HMP 1.2

Defect Number	PTR Number	SU No.	Product or Component	Description
IPY00033948	--	18	HMP	dlstop does not stop all Dialogic processes if the user's default shell is csh.
IPY00033758	--	18	HMP	teeloger process is created each time ct_intel restart is executed.
IPY00032236	36062	18	Kernel	Kernel fails with a QVERIFY FAILED in lbrac.c 220 after approximately 25 days.
IPY00032235	36405	18	Kernel	Kernel fails with a qerror_killtask AND ERROR CODE 0X41107 after approximately 25 days.
IPY00031503	36781	18	HMP	Changing the ethernet interface name from the default of ethX produces an error (e.g., "Kernel failed to interpret the license key, boardNumber=1, processor=CP") on Dialogic system services start up.
IPY00032236	36062	14	Kernel	Kernel fails with a QVERIFY FAILED on the lbrac.c file after approximately 25 days.
IPY00032235	36405	14	Kernel	Kernel fails with a QERROR_KILLTASK and error code 0x41107 after approximately 25 days.
IPY00023802	36484	14	IP Host	Multiple channels sending RTP data to a same IP address and same RTP port cause the channels to hang.
IPY00009054	34875	9	HMP	Because the maximum number of instances in the PCD file was set too low, a killtask (and eventually a blue screen) occured when trying to download the 240R-240V-120E-240C-240S-64F SKU.
IPY00010596	34758	9	HMP	The free default single channel SKU should be 1r1v0e0c0s0f_ver.xml, not 1r1v1e1c1s1f_ver.xml.
IPY00009813	34638	9	Voice	Adding a tone using the dx_addtone( ) function, followed by the dx_distone( ) function on a high density channels (for example, 240) results in some tones failing.
IPY00030253	34363	9	Voice	HMP conferencing uses a phase reversal algorithm to improve audio quality in conferencing. However, typical PESQ tests do not recognize this and report low PESQ. A parameter has been added that allows the phase reversal algorithm to be disabled so PESQ tests will not fail.
IPY00001238	34148	6	HMP	The function dx_open() takes up to 10 seconds to complete when running as non-root user. The function returns successful after those 10 seconds. When running as root, there is no such delay.
IPY00030251	33919	6	HMP	Upon running different consecutive runs of the same application, hmpperf, in different configurations, the 90th run always fails.

Table 1. Issues Resolved in HMP 1.2 (Continued)

Defect Number	PTR Number	SU No.	Product or Component	Description
IPY00009204	33894	6	CSP/Voice	During heavy load, while running CSP and voice together, the HMP system may miss an event and have a channel stay busy.
IPY00009082	33762	6	Global Call	Segmentation fault in gc_Start() when more than 1024 files have been opened prior to calling gc_Start(). If 1024 files are opened after gc_Start() and after opening the devices, no calls can be established.
IPY00009801	33644	6	HMP	The interface does not indicate if it is necessary to reboot after activating a license and uninstalling the software. If there is a reboot message in the software script, it scrolls too quickly for users to see.
IPY00009304	33638	6	HMP	Difficult to see the "success" messages on the screen because several error messages run into the "success' message.
IPY00011176	33609	6	Install	HMP Linux 1.2 software will not install correctly on a system that does not have compat-libstdc++-7.3-2.96.123.i386.rpm installed. The following error message is displayed when the HMP configuration script is run: `fcdgen: error while loading shared libraries: libstdc++-libc6.2-2.so.3: cannot open shared object file: No such file or directory.`
IPY00011219	33449	6	Install	If you try to use the install script a second time to install a package that has already been installed, disk space check will fail and the install script will terminate and exit. The disk space check failure does not necessarily mean the system is low on disk space.
IPY00011186	33401	6	Global Call	Applications running H.323 under Global Call with the default list of coders (don't care) will see error messages on the console.
IPY00011189	33395	6	License Manager	The LicenseManager program includes the -l option, which reads and executes commands from the LicenseManager.ini file. The LicenseManager expects this file to reside in /usr/dialogic/ini, but the ini subdirectory does not currently exist in HMP LInux 1.2.
IPY00011191	33393	6	CSP	Segmentation Fault occurs in the IP Media Server demo if the CSP option is chosen and a non-CSP licensed is installed.
IPY00011217	33386	6	Global Call	Applications running H.323 under Global Call may get GCEV_DISCONNECTED for an outbound call with the error message: IPEC_Q931TransportError(0xE75) This error is caused by socket errors.
IPY00011248	33380	6	CSP	The ec_stream() and ec_reciottdata() functions do not return errors back to the application when an unsupported recorded format is used.

Table 1. Issues Resolved in HMP 1.2 (Continued)

Defect Number	PTR Number	SU No.	Product or Component	Description
IPY00011249	33376	6	Install	During the install, instead of entering 1 to install the HMP software, if a 1` entered. (the 1 and the key to the left of the 1 key), the install prints out some error messages and then installs only two of the packages (OOC and DMDEV).
IPY00011233	33363	6	HMP	Merc Stream Rcv Disconnect(): unexpected Drv. Msg C00013 prints to the screen while running tests.
IPY00024459	33325	6	CSP	If the ec_stream() function is used with FILE_FORMAT_VOX, DATA_FORMAT_PCM and BitsPerSample=16, the resulting recorded file is in DATA_FORMAT_MULAW.
IPY00011174	33312	6	CSP	CPU usage rises to 100% when testing continuous speech processing on more than 96 channels with echo cancellation enabled.
IPY00011129	33268	6	Global Call	Global Call library does not handle error scenario when an unmatched callIdentity is received in H.450.2 call xfer.
IPY00011283	32818		Global Call	A segfault occurs in gc_Stop() regardless of whether RTF log files are deleted.
IPY00011284	32817	6	Global Call	A segfault occurs in gc_GetMetaEvent() after gc_AcceptCall() is issued. Debugging shows the event being retrieved is GCEV_EXTENSIONCMPLT. This problem seems to occur with loads that do not use low bit rate coders, and can only be reproduced if RTF log files are deleted prior to running the application.
IPY00011329	32518	6	Global Call	Application may hang after calling gc_Close() function.
IPY00011489	32514	6	HMP	You may receive a compiler warning in ctinfo.h about an undefined token after the #endif directive. This warning will have no impact on your application and may be safely ignored.

Table 2. Known Issues in HMP 1.2

Defect Number	PTR Number	Product or Component	Description
IPY00028330	34473	Demo	When NetMeeting on Windows XP is used to call into IPMS demo on HMP, the call is established, but digit press is not recognized by the demo side. ***Workaround:*** None
IPY00028493	34440	Demo	When a voice mail message is successfully recorded and then listened to, the option to either save the message or delete it does not work. ***Workaround:*** Recompile the demo application. From the following path: /usr/dialogic/demos/ip_demos/IPMediaServer, enter the command: >make clean followed by: >make all
IPY00006505	34425	CSP	When running CSP applications with 120 or more channels, a stuck channel may occur occasionally when the TDX_BARGEIN event is missed during CSP operations. ***Workaround:*** Stop all cron jobs while running high channel density CSP applications.
IPY00010138	33783	CSP	An HMP system failure may occur in an application with channel densities above 40 running on a computer with 256K cache or less. ***Workaround:*** For channel densities above 40, use a computer with Intel® Pentium® 4 processors that have greater than 256K cache (default for newer computers is a cache size greater than 512K). The size of the cache on the machine can be determined by looking in the file /proc/cpuinfo.

2.2 Restrictions and Limitations

Known restrictions and limitations in this release can be categorized as follows:

- General
- Voice
- Fax
- Global Call, IP, and IPML
- Voice

General

- High I/O activity (for example, updatedb) during heavy HMP activity may cause an increase in error rates, such as degraded digit detection and voice quality.

- DM3Stderr and DebugAngel diagnostics tools are not supported in this release. As a way to debug DM3 firmware, all firmware prints are sent to /var/log/messages by default. They can be sent to stdout when the boot kernel is run in non-daemon mode. You must ensure that syslogger is enabled and running. To check, use “ps -ef | grep syslogger”. If it is not running, you can start it using the Linux setup utility.
- If you run an application as root and then switch to a non-privileged user, any log files created in the first run will probably be read-only for the non-privileged user.
- For users who do not use DNS or have a system with no DNS entry, the /etc/hosts file must contain an entry with the machine’s hostname and IP address. The hostname should only appear once in the file.
- The HMP system does not operate with board level products and system software releases installed on the same machine.
- All cron jobs should be scheduled for off-peak hours to avoid performance issues.

Conferencing

- The **dcb_setbrdparm()** function fails when attempting to set MSG_ALGORITHM to ALGO_LOUD and ATDV_ERRMSGP returns "Bad global parameter value". Do not set this parameter. By default, the algorithm uses the Loudest Talker.

Fax

- For fax applications, the header file *srllib.h* must appear before the header file *faxlib.h* in the #include directive.

Global Call, IP, and IPML

- Only IP-specific Global Call features will be provided as described in the *Global Call IP for Host Media Processing Technology Guide* (05-2239-003).
- G.711 µ-Law and A-Law IP Encoding/Decoding only support 64 Kbps.
- If a call is made to HMP using NetMeeting* from a machine that does not have a sound card, the coder negotiation will fail and the call will be disconnected before a GCEV_ANSWERED event is generated.
- If a call is made to HMP using NetMeeting, the "secure outgoing calls" option must not be selected.
- Global Call applications using T.38 should call **gc_SetUserInfo(IPPARM_T38_CONNECT)** after GCEV_OFFERED, but before the next GC function call, for T.38 only calls. Otherwise, the call will fail and the application will get a GCEV_TASKFAIL event.
- The **ipm_SendRFC2833SignalToIP()** function is not supported on HMP. The application cannot use this API to generate RFC2833 digits on a call. Use the voice device to dial digits and set up RFC2833 transfer mode in a call.
- Host applications should always clean up resources before exiting by using one of the following methods:
  - If using Global Call call control, the application should call **gc_DropCall()** followed by **gc_ReleaseCall()**, or use **gc_ResetLineDev()**.

- If using a non-Global Call call control stack, the application should stop RTP sessions by using **ipm_Stop()**.
- Using the **dlstop** and **dlstart** commands to stop and then start HMP.

- An application should not open more than 1024 files in a process. If an application needs to use many files for media operations, the files should be opened and closed as needed instead of keeping them open throughout the process.

Voice

- The **dx_reciottdata()** and **dx_playiottdata()** functions do not support recording or playing in WAVE format to/from memory; only VOX format.
- Global Tone Detection (GTD) does not support detecting user defined tones as digits via the digit queue; the tones are only detected as tone events via the event queue.
- For dual tone definition, the frequency deviation that is defined in the tone template for each frequency must be not less than ± 30 Hz.
- For single tone definition, the frequency deviation that is defined in the tone template for each frequency must be not less than ± 30 Hz.
- The number of tone templates which can be added to a voice device and enabled for detection is limited. The maximum number of events for each instance is 10.
- The **dx_dial()** function only detects CED tones for CR_FAXTONE event. CNG tones cannot be detected.
- The **dx_playtoneEx()** function will not terminate after the time specified by the DX_MAXTIME termination parameter has elapsed.
- DTMF digits not processed by the user application with **dx_getdig()** or other means remain in the device after it is closed with **dx_close()**, and remain with the device even after the application terminates. To ensure that the buffer is empty, clear the digit buffer using **dx_clrdigbuf()**.
- The **dx_getctinfo()** and **fx_getctinfo()** functions return incorrect values for Device Family and Bus Mode.
- G.711 10 ms is not supported at densities above 120 channels.
- G.726 is not supported at densities above 120 channels.

Demos

- Demos are intended to be run using a maximum of 4 channels. Also, the channel numbers selected for the demo should be 120 or lower. Using channel numbers higher than 120 could cause the demo to fail.

Resolved Restrictions and Limitations

The following Restrictions and Limitations have been resolved:

Host-based H.323 Protocol Stack

Global Call applications that use the host-based H.323 protocol stack may generate the following error message in the gc_h3r.log file if the application enables the stack to send the PROCEEDING message automatically.

```
! 22:22:19.393 ! M_SIGNAL ! L_ERROR  !  1 ! << SIGNAL::sendProcceding: RV
cmCallSendCallProceeding Failed : [-996]
```

This error message can be ignored. The PROCEEDING message is actually sent.

2.3 Compatibility Notes

Echo Cancellation With Continuous Speech Processing

For HMP Continuous Speech Processing (CSP), the Echo Cancellation parameter default value is set to OFF (echo cancellation disabled). To conserve CPU usage/MIPs, the application should keep this parameter value set to OFF.

For additional information about CSP, refer to the *Continuous Speech Processing API for Linux and Windows Library Reference* and the *Continuous Speech Processing API for Linux and Windows Programming Guide*.

Echo Cancellation With Conferencing

For HMP conferencing, the Echo Cancellation parameter default value is set to OFF (echo cancellation disabled) for compatibility with DCB applications. This parameter must not be turned ON by the application. Enabling echo cancellation causes poor conference audio quality.

For additional information about Conferencing, refer to the *Audio Conferencing API for Linux and Windows Operating Systems Library Reference* and the *Audio Conferencing API for Linux and Windows Operating Systems Programming Guide*.

Configuring UDP/RTP Port Range

The HMP system currently defaults to UDP/RTP ports 6000 through 6100 and UDP ports 49152 through 49xxx for RTP streaming, where 49xxx = 49152 + twice the maximum number of channels purchased under the licensing agreement. For example, if 120 channels were purchased, 49xxx would be 49392.

If the UDP/RTP port range used by the HMP system conflicts with other RTP service, the following procedure describes how to configure HMP to use a different UDP/RTP port range:

1. Stop the system service.

2. Locate the .config file in the */usr/dialogic/data* directory that matches the FCD/PCD files associated with your licensed configuration.

3. Using a text editor, open the .config file.

4. Locate the [IPVSC] section in the .config file.

5. In the [IPVSC] section, locate the line

   ```
    setparm 0x4005, 49512   !set the rtpPortBase on IPVSC
   ```

   The number 49512 is the default value for this parameter. You may change the beginning of the UDP/RTP port range by first editing this value and saving the .config file.

6. After you have saved the .config file with the new UDP/RTP port value, open the Command Prompt window.

7. From the Command Prompt, change the directory to */usr/dialogic/data*.

8. Execute fcdgen as follows:

   ```
   /usr/dialogic/bin/fcdgen -f <input filename>.config -o <output
   filename>.fcd
   ```

   The resulting FCD file is created in the */usr/dialogic/data* directory. If the -o option is omitted from the command, the default output FCD file will have the same filename as the user-modified input .config file, but with an .fcd extension.

9. Restart the system service to download the new configuration to HMP.

Configuring T.38 Service Port Range

The HMP system currently defaults to port 6000 as the starting UDP/RTP port for the T.38 service port. If the T.38 service port range used by the HMP system conflicts with other RTP service, the following procedure describes how to configure HMP to use a different T.38 service port range:

1. Stop the system service.

2. Locate the .config file in the */usr/dialogic/data* directory that matches the FCD/PCD files associated with your licensed configuration.

3. Using a text editor, open the .config file.

4. Locate the [0xe] section in the .config file.

5. In the [0xe] section, locate the line

   ```
   setparm 0x4c21, 6000 ! QFC3_PrmIPRxPortBase (QFC3_PrmLocalUDPPortBase)
   ```

The number 6000 is the default value for this parameter. You may change the beginning of the T.38 service port range by first editing this value and saving the .config file.

6. After you have saved the .config file with the new T.38 service port value, open the Command Prompt window.

7. From the Command Prompt, change the directory to */usr/dialogic/data*.

8. Execute fcdgen as follows:

   ```
   /usr/dialogic/bin/fcdgen -f <input filename>.config -o <output
   filename>.fcd
   ```

   The resulting FCD file is created in the */usr/dialogic/data* directory. If the -o option is omitted from the command, the default output FCD file will have the same filename as the user-modified input .config file, but with an .fcd extension.

9. Restart the system service to download the new configuration to HMP.

H.323 Coder Negotiation with Third Party Stacks

Use caution when restricting coder frame sizes or frames per packet while communicating with third-party H.323 stacks. The IPT H.323 protocol stack uses both coder type and frames per packet as part of the algorithm to determine a successful Tx/Rx media match. Restricting coder frame sizes can cause Fast Start calls to fallback to Slow Start or coder negotiation to fail. See the following example.

Example:

> A Global Call application configures a channel for G.711 A Law with a frame size of 10 milliseconds. A third-party H.323 stack, which is configured for G.711 A-Law with a frame size of 30 milliseconds, initiates a call to the application. The IPT H.323 protocol stack will use 10 milliseconds as an upper limit for both the Tx and Rx media directions. Even though both sides support the same coder, the frame size discrepancy can cause the coder negotiation to fail, resulting in a GCEV_DISCONNECTED event.

H.323 T.38 HMP Interoperability with the Intel NetStructure® PBX-IP Media Gateway

HMP has not been tested with the current version of the Intel NetStructure PBX-IP Media Gateway (Version 4.0 Service Update 2) when using H.323 T.38 Fax.

SIP Call Control

If the remote site does not respond to an outgoing INVITE sent from HMP, the gc_MakeCall() function will time out after 32 seconds and generate a GCEV-DISCONNECTED event. In this scenario, if the application attempts to drop the call before the 32 second timeout is reached, a CANCEL will be sent by HMP to the remote site. If there is no response by the remote site to the CANCEL, there will be an additional 32 second timeout, at the end of which, a GCEV_DISCONNECTED event will be reported.

Supervised Transfer

H.450.2 Supervised Transfer has not been tested with Cisco 5300 running IOS 12.3(8).

2.4 Operating System Notes

Using Multiple NICs Under Heavy Loads

When using multiple network interface cards (NICs) under heavy loads, kernel assertions may occur.

Symptom: Calls may fail with the following message displayed:

```
kernel: KERNEL: assertion (flags & MSG_PEEK) > failed at net/ipv4/tcp.c
(1284)
```

Solutions:

Resolution 1 - If you have two NICs on one segment, turn off **Forwarding** on at least one of the NICs.

```
Eth0: 192.168.66.21
Eth2: 192.168.66.77

net.ipv4.conf.eth0.forwarding = 0
net.ipv4.conf.eth2.forwarding = 0
```

Resolution 2 - Update the following system parameters as shown:

```
sysctl -w net.ipv4.tcp_max_syn_backlog=2048
sysctl -w net.ipv4.route.max_size=1048576
sysctl -w net.ipv4.route.gc_thresh=65536
```

Elvtune Using IOCTLS

Elvtune using IOCTLS is now deprecated. To tune the IO scheduler, use the files exported to the sysfs directory.

```
/sys/block/<device>/queue/iosched
```

Documentation Updates 3

The documentation updates are divided into the following sections, which correspond to the top level categories used in the online documentation navigation page:

3.1 Release Documentation

This section contains updates to the following documents:

- Host Media Processing Software Release 1.2 for Linux Release Guide

3.1.1 Host Media Processing Software Release 1.2 for Linux Release Guide

In **Chapter 2, "System Requirements"**, the following information applies to the **Basic Hardware Requirements** section:

- The footnote in Table 1 should read: "‡ = Up to a maximum of 120 low bit rate coder sessions."

In **Chapter 2, "System Requirements"**, the following information applies to the **Software Requirements** section:

The software requirements for this release are:

- Red Hat Enterprise Linux Advanced Server Release 3 with Update 1 or Update (with kernel version 2.6.8.1 provided by Intel)

  or

- SUSE Linux Enterprise Server 9.2 (includes kernel version 2.6.8-24)
- Supported compilers: GCC Version 3.2.3

In **Chapter 2, “System Requirements”**, the following section is added:

2.4 Running a Demo Using a Single Channel

The default channel license provided with the HMP Linux 1.2 software relies on the correctness of the */etc/hosts* file. Invalid entries in this file will render the HMP product non-functional with respect to the default channel license. For example:

1.	127.0.0.1	myhostname	<< **incorrect**
2.	127.0.0.1	localhost	<< **incorrect**
	192.168.124.124	localhost	
3.	127.0.0.1	localhost	<< **correct**
	192.168.124.124	myhostname	

In **Chapter 3, “ Release Features”**, the following information applies to the **Configurations Tested** section:

- Current testing of 240 channels is performed with G.711 20 ms and 30 ms frames only.
- Current testing of 240 channels is performed using the Intel 8254x-based gigabit Ethernet solution. Other network interface cards may not provide optimum performance. However, 8241, 8245, 8246, and 8247-based gigabit cards are also expected to perform well. These cards provide superior performance by:
  - Minimizing the number of interrupts needed relative to work accomplished.
  - Reducing the frequency of all interrupts by setting the “Interrupt Throttling Register” to the appropriate value.
  - Accepting multiple receive packets before signaling an interrupt.
  - Eliminating (or at least reducing) the need for interrupts on transmit.
  - Minimizing the processing overhead associated with each interrupt.

The following procedure for obtaining Update 1 to Red Hat Enterprise Linux Advanced Server 3.0 applies to **Section 2.2, “Software Requirements”** of **Chapter 2, “System Requirements”**:

1. If you do not already have a login, create a login and subscribe to the Red Hat network at *https://rhn.redhat.com/newlogin/create_personal.pxt*. You will then need to enter the Red Hat product ID.

2. Sign in to the Red Hat Network at *https://rhn.redhat.com/index.pxt*.

3. Select **Channels** from the tab selections at the top of the page.

4. Select **Red Hat Enterprise Linux AS** from the Channel Name list.

5, Select **Downloads** from the menu selections.

6. After reading and accepting the License Agreement, follow the instructions provided on the Web page to download the four ISO Binary Discs that comprise Update 1. Also, refer to the Red Hat Installation Manual for additional information.

In **Chapter 3, " Release Features"**, the following information applies to the **Configurations Tested** section:

- Table 2 has been changed as follows to reflect the support of 120 Enhanced RTP resources.

Configuration	RTP	Enhanced RTP	Voice	Conferencing (DCB)	Fax	Speech
IVR (G.711)	120	0	120	0	0	0
IVR (LBR)	120	120	120	0	0	0
IVR (G.711)	240	0	240	0	0	0
Unified Messaging	60	12	60	60	6	60
Unified Messaging	96	60	96	24	8	96
Unified Messaging	200	0	200	0	14	140
Gateway (SIP)	23	11	23	0	6	23
Gateway (H.323)	23	11	23	0	6	23
Conferencing (DCB)	100	60	50	100	0	0
Conferencing (DCB)	240	0	100	240	0	0

In **Chapter 4, "Development Software"**, the following apply:

- The bullet describing "Active Talker" in the **Features** section should read: "Active talker indication to determine which conferees in any given conference are currently talking."
- The only Audio Conferencing API function not supported is: dcb_GetAtiBitsEx()

In **Chapter 5, "Release Documentation"**, the following apply:

- The correct part number for the *IP Media Library API for Host Media Processing Library Reference* is 05-2257-003.
- The correct part number for the *IP Media Library API for Host Media Processing Programming Guide* is 05-2330-001.
- The correct part number for the *Voice API for Host Media Processing Programming Guide* is 05-2332-001.

- The IP Media Server (Global Call) Demo Guide (P/N 05-2065-001) has been replaced by the IP Media Server for Host Media Processing Demo Guide (P/N 05-2389-001).
- The correct title for the Diagnostics Guide is *Intel NetSructure Host Media Processing Diagnostics Guide*.

3.2 Installation Documentation

This section contains updates to the following documents:

- Host Media Processing Software Release 1.2 for Linux Installation Guide

3.2.1 Host Media Processing Software Release 1.2 for Linux Installation Guide

The following information applies:

- In **Section 2.1 "Assumptions and Prerequisites"** of **Chapter 2, "Installing the Software"**, the correct version of the Intel-provided kernel is 2.6.8.1.
- After installing or uninstalling the HMP Linux 1.2 software, it is not necessary to reboot the system. You will, however, need to set the environment variables after installing the software. To set the environment variables, enter: *source /etc/profile.d/ct_intel.sh*

  or, alternatively: logout and then login.
  (IPY00008512 = PTR 33848)

- In Step 4 of the procedure for uninstalling the HMP Linux software in **Chapter 3, "Uninstalling the Software"**, you must use the following procedure to start the License Manager:

1. If the system services are started, enter the *hmpconfig* command; if the system services are stopped, enter the *config.sh* command (both are located in the /usr/dialogic/bin directory) to start the configuration tool.

2. If using *config.sh*, then once inside the configuration tool, press Enter to start *hmpconfig* and display the License Manager main menu.
   (IPY00009997 = PTR 33828)

3.3 Operating Software Documentation

This section contains updates to the following documents:

- Host Media Processing Software Release 1.2 for Linux License Manager Administration Guide
- Host Media Processing Diagnostics Guide

3.3.1 Host Media Processing Software Release 1.2 for Linux License Manager Administration Guide

In all instances, the correct procedure for starting the License Manager is:

1. Enter the config.sh command (located in the /usr/dialogic/bin directory) to start the configuration tool.

2. Once inside the configuration tool, press Enter to start hmpconfig and display the License Manager main menu.
   (IPY00009997 = PTR 33828)

The procedures for changing or removing the HMP IP address should include an initial step of stopping HMP services by running the `dlstop` script and a final step of restarting the HMP services by running the `dlstart` script. This affects the following procedures:

- In **Chapter 9, "Changing the HMP IP Address"**, add these steps to the **Using the Menu Version of the License Manager** and the **Using the CLI Version of the Licensing Manager** procedures.
- In **Chapter 10, "Removing a User-Selected HMP IP Address"**, add these steps to the **Using the Menu Version of the License Manager** and **Using the CLI Version of the Licensing Manager** procedures.
  (IPY00010180 = PTR 33850)

The information in **Chapter 11, "Stopping and Restarting the HMP Services"** should be as follows:

This chapter describes how to manually stop and restart the HMP services.

If you need to stop and restart the HMP services:

- Run the `dlstop` script to stop the HMP services
- Run the `dlstart` script to restart the HMP services

Notes:1. Before you run these scripts, Linux must be started with the kernel supported for this release (refer to the Release Guide).

2. The HMP services are started automatically when you reboot your system.

3.3.2 Host Media Processing Diagnostics Guide

The following changes and corrections apply:

- Disregard **Chapter 4, "DM3StdErr Reference"** and all references to the Dm3StdErr tool because this tool is not supported by the HMP Linux 1.2 software.
- In **Chapter 5, "Getver Reference,"** the filename in the example should be ssp.mlm instead of qvs_t1.mlm.
- Disregard **Chapter 7, "QError Reference"** because the QError tool is obsolete and will be removed from future versions of the HMP Linux software.

3.4 Development Software Documentation

This section contains updates to the following documents:

- Audio Conferencing API for Linux and Windows Operating Systems Library Reference
- Audio Conferencing API for Linux Operating Systems Programming Guide
- Continuous Speech Processing API for Linux and Windows Operating Systems Library Reference
- Global Call IP for Host Media Processing Technology Guide
- Voice API for Host Media Processing Library Reference
- Voice API for Host Media Processing Programming Guide

3.4.1 Audio Conferencing API for Linux and Windows Operating Systems Library Reference

Update to **dcb_getbrdparm( )** and **dcb_setbrdparm( )** functions (IPY00006584 = PTR 36199)

Changed the description of the MSG_ACTID parameter to indicate that it enables/disables active talker identification (or notification) and not the active talker feature itself. Replace the MSG_ACTID parameter with the following:

Enables or disables Active Talker Identification (or Notification). Possible values are ACTID_ON or ACTID_OFF. ACTID_ON is the default. This parameter does not enable or disable the active talker *feature*, which is always enabled. It only disables the *notification* to the application program. The active talker *feature* sums the 3 most active talkers in a conference, so that the conversation doesn't get drowned out when too many people talk at once. Active talker *notification* provides data on active talkers through the **dcb_gettalkers( )** and **dcb_GetAtiBitsEx( )** functions, which can be used by an application program to identify active talkers; for example, to provide a visual display highlighting the active talkers in a conference. Active talkers are determined by their loudness; i.e., the strength of their "non-silence" energy.

Note: In some cases, it is desirable to inactivate the active talker feature, such as for a background music application program. Although you cannot directly disable the active talker *feature*, you can set the noise level threshhold by which signals are recognized as either speech or noise.

3.4.2 Audio Conferencing API for Linux Operating Systems Programming Guide

Update to Section 6.2, "Initialization of DM3 Board Parameters" and to "Active Talker" chapter (IPY00006584 = PTR 36199)

Changed the description of the MSG_ACTID parameter to indicate that it enables/disables active talker identification (or notification) and not the active talker feature itself. Replace the MSG_ACTID parameter with the following:

MSG_ACTID (Active Talker Identification)

Enables or disables Active Talker Identification (or Notification). Possible values are ACTID_ON or ACTID_OFF. ACTID_ON is the default. This parameter does not enable or disable the active talker *feature*, which is always enabled. It only disables the *notification* to the application program. The active talker *feature* sums the 3 most active talkers in a conference, so that the conversation doesn't get drowned out when too many people talk at once. Active talker *notification* provides data on active talkers through the**dcb_gettalkers( )** and **dcb_GetAtiBitsEx( )** functions, which can be used by an application program to identify active talkers; for example, to provide a visual display highlighting the active talkers in a conference. Active talkers are determined by their loudness; i.e., the strength of their "non-silence" energy.

Note: In some cases, it is desirable to inactivate the active talker feature, such as for a background music application program. Although you cannot directly disable the active talker *feature*, you can set the noise level threshhold by which signals are recognized as either speech or noise.

3.4.3 Continuous Speech Processing API for Linux and Windows Operating Systems Library Reference

For more information on DV_TPT termination conditions when using **ec_reciottdata()** and **ec_stream()** functions, see the DV_TPT data structure description in the *Voice API for Host Media Processing Library Reference*.
(IPY00010008 = PTR 33269)

The code examples for the functions **ec_reciottdata()** and **ec_stream()** specify a data format that is not valid on HMP. The example code should be revised as follows:

Original

```
..
xpb.wFileFormat = FILE_FORMAT_VOX;
xpb.wDataFormat = DATA_FORMAT_PCM;
xpb.nSamplesPerSec = DRT_8KHZ;
xpb.wBitsPerSample = 8;
..
```

Revised

```
..
xpb.wFileFormat = FILE_FORMAT_VOX;
xpb.wDataFormat = DATA_FORMAT_MULAW;
xpb.nSamplesPerSec = DRT_8KHZ;
xpb.wBitsPerSample = 8;
..
```

(IPY00009264 = PTR 33379)

3.4.4 Continuous Speech Processing API for Linux and Windows Operating Systems Programming Guide

There are currently no updates to this document.

3.4.5 Device Management API for Windows and Linux Operating Systems Library Reference

There are currently no updates to this document.

3.4.6 Fax Software Reference for Linux and Windows

There are currently no updates to this document.

3.4.7 Global Call API for Host Media Processing Library Reference

There are currently no updates to this document.

3.4.8 Global Call API for Host Media Processing on Linux Programming Guide

There are currently no updates to this document.

3.4.9 Global Call IP for Host Media Processing Technology Guide

The following information applies:

- Update Section 7.2.16, "gc_MakeCall( ) Variances for IP" (IPY00029956 = PTR 36646) the last paragraph before Section 7.2.16.1 (page 173) as follows:

  When using SIP, if the remote side does not send a final response to an outgoing INVITE (sent by the call control library) within 64 seconds, the gc_MakeCall( ) function times out and the library generates a GCEV_DISCONNECTED event to the application. If the application attempts to drop the call before the 64 second timeout is reached, the library's behavior depends on whether a provisional response was received. When no provisional response was received before the application cancels the call, the library cleans up the call immediately. But if a provisional response was received before the application attempts to cancel the call, the library sends a CANCEL to the remote endpoint and generates a GCEV_DROPCALL to the application after it receives the 200OK response to the CANCEL and a 487RequestTerminated response for the original INVITE, or when an additional 32-second timeout expires.
- On the reference page for the IP_H221NONSTANDARD data structure (page 241), the descriptions of the three data fields are updated as follows:

country_code

The country code. Range: 0 to 255; any value x>255 is treated as x%256.

extension

The extension number. Range: 0 to 255; any value x>255 is treated as x%256.

manufacturer_code

The manufacturer code. Range: 0 to 65535; any value x>65535 is treated as x%65636.

- In **Section 4.16, Registration**, on page 113, the fourth bullet item in the second list ["Once an application is registered, if it wishes to handle calls without the registration protocol (that is, return to the same node as before registration), it can simply deregister."] is incorrect and should be ignored.
- In **Section 7.2.26, gc_Start() Variances for IP,** the note following the #define IP_CFG_MAX_AVAILABLE_CALLS applies to both HMP Windows and Linux versions, and should read:

  Note: Do not use IP_CFG_MAX_AVAILABLE_CALLS with applications running on HMP, since IP_CFG_MAX_AVAILABLE _CALLS will intitialize the stack for 2016 channels. This is an inefficient use of system resources, resulting in a lengthy initialization time and substantial memory use.
- In **Section 7.2.17, gc_OpenEx() Variancesfor IP**, the following note should be included:

  Note: Applications should avoid closing and re-opening devices multiple times. Board devices and channel devices should be opened during intialization and should remain open for the duration of the application.

3.4.10 IP Media Library API for Host Media Processing Library Reference

There are currently no updates to this document.

3.4.11 IP Media Library API for Host Media Processing Programming Guide

There are currently no updates to this document.

3.4.12 Standard Runtime Library API for Linux and Windows Operating Systems Library Reference

There are currently no updates to this document.

3.4.13 Standard Runtime Library API for Linux Operating Systems Programming Guide

There are currently no updates to this document.

3.4.14 Voice API for Host Media Processing Library Reference

The **dx_mreciottdata( )** function has been added to support the Transaction Record feature. See the following for information about this function.

dx_mreciottdata( )

Name:	dx_mreciottdata (devd, iotp, tptp, xpb, mode, sc_tsinfop)	
Inputs:	int devd	• valid channel device handle
	DX_IOTT *iotp	• pointer to I/O transfer table
	DV_TPT *tptp	• pointer to termination control block
	DX_XPB *xpb	• pointer to I/O transfer parameter block
	USHORT *mode	• switch to set audible tone, or DTMF termination
	SC_TSINFO *sc_tsinfop	• pointer to time slot information structure
Returns:	0 success -1 error return code	
Includes:	srllib.h dxxxlib.h	
Category:	I/O	

Mode: asynchronous or synchronous

Description

The **dx_mreciottdata( )** function records voice data from two TDM bus time slots. The data may be recorded to a combination of data files, memory or custom devices.

This function is used for the transaction record feature, which allows you to record two TDM bus time slots from a single channel. Voice activity on two channels can be summed and stored in a single file, device, and/or memory.

Parameter	Description
devd	specifies the valid channel device handle on which the recording is to occur. The channel descriptor may be that associated with either of the two TDM bus transmit time slots or a third device also connected to the TDM bus.
iotp	points to the I/O Transfer Table Structure, DX_IOTT, which specifies the order of recording and the location of voice data. For more information on this structure, see the DX_IOTT data structure.
tptp	points to the Termination Parameter Table Structure, DV_TPT, which specifies the termination conditions for recording. For more information on this structure, see the DV_TPT data structure.
xpb	points to a DX_XPB structure, which specifies the file format, data format, sampling rate, and resolution for I/O data transfer. For more information on this structure, see the DX_XPB data structure.
mode	specifies the attributes of the recording mode. One or more of the following values can be specified: • 0 – standard record mode • RM_TONE – transmit a 200 msec tone before initiating record
sc_tsinfop	points to the SC_TSINFO data structure and specifies the TDM bus transmit time slot values of the two time slots being recorded. In the SC_TSINFO structure, **sc_numts** should be set to 2 for channel recording and **sc_tsarrayp** should point to an array of two long integers, specifying the two TDM bus transmit time slots from which to record.

Note: When using RM_TONE bit for tone-initiated record, each time slot must be “listening” to the transmit time slot of the recording channel; the alert tone can only be transmitted on the recording channel’s transmit time slot.

After **dx_mreciottdata( )** is called, recording continues until one of the following occurs:

- **dx_stopch( )** is called on the channel whose device handle is specified in the **devd** parameter
- the data requirements specified in the DX_IOTT structure are fulfilled
- one of the conditions for termination specified in the DV_TPT structure is satisfied

Cautions

- All files specified in the DX_IOTT structure are of the file format specified in DC_XPB.
- All files recorded will have the same data encoding and rate as DC_XPB.

- When recording VOX files, the data format is specified in DC_XPB rather than through the **dx_setparm( )** function.
- Voice data files that are specified in the DX_IOTT structure must be opened with the O_BINARY flag.
- If both time slots transmit a DTMF digit at the same time, the recording will contain an unintelligible result.
- Since this function uses **dx_listen( )** to connect the channel to the first specified time slot, any error returned from **dx_listen( )** will terminate the function with the error indicated.
- This function connects the channel to the time slot specified in the SC_TSINFO data structure **sc_tsarrayp[0]** field and remains connected after the function has completed. Both **sc_tsarrayp[0]** and **sc_tsarrayp[1]** must be within the range allowed in SC_TSINFO. No checking is done to verify that **sc_tsarrayp[0]** or **sc_tsarrayp[1]** has been connected to a valid channel.
- Upon termination of the **dx_mreciottdata( )** function, the recording channel continues to listen to the first time slot (pointed to by **sc_tsarray[0]**).
- The application should check for a TDX_RECORD event with T_STOP event data after executing a **dx_stopch( )** function during normal and transaction recording. This will ensure that all data is written to the disk.
- When using **dx_mreciottdata( )** and a dial tone is present on one of the time slots, digits will not be detected until dial tone is no longer present. This is because the DSP cannot determine the difference between dial tone and DTMF tones.
- Tone termination conditions such as DTMF and TONE apply only to the primary input of the function; that is, the TDM time slot specified in the SC_TSINFO data structure **sc_tsarrayp[0]** field.

■ Errors

If the function returns -1, use the 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_BADDEV
Invalid device handle

EDX_BADIOTT
Invalid DX_IOTT entry

EDX_BADPARM
Invalid parameter passed

EDX_BADTPT
Invalid DV_TPT entry

EDX_BUSY
Busy executing I/O function

EDX_SYSTEM
Error from operating system

■ Example

The following example is for Linux applications.

```
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <srllib.h>
#include <dxxxlib.h>

#define MAXLEN 10000

main()
{
   int devh1, devh2, devh3;
   short fd;
   DV_TPT tpt;
   DX_IOTT iott[2];
   DX_XPB xpb;
   SC_TSINFO tsinfo;
   long scts;
   long tslots[32];
   char basebufp[MAXLEN];

   /* open two voice channels */
   if ((devh1 = dx_open("dxxxB1C1", NULL)) == -1) {
      printf("Could not open dxxxB1C1\n");
      exit (1);
   }

   if ((devh2 = dx_open("dxxxB1C2", NULL)) == -1) {
      printf("Could not open dxxxB1C2\n");
      exit (1);
   }

   if ((devh3 = dx_open("dxxxB1C3", NULL)) == -1) {
      printf("Could not open dxxxB1C2\n");
      exit (1);
   }

   if ((fd = open("file.vox", O_CREAT | O_RDWR,  0666)) == -1){
      printf("File open error\n");
      exit (1);
   }

   /*
    * Get channels' external time slots
    * and fill in tslots[] array
    */

   tsinfo.sc_numts = 1;
   tsinfo.sc_tsarrayp = &scts;

   if (dx_getxmitslot (devh1, &tsinfo) == -1 )
      { /* Handle error */ }
   tslots[0] = scts;
   if (dx_getxmitslot (devh2, &tsinfo) == -1 )
      { /* Handle error */ }
   tslots[1] = scts;

   /* Set up SC_TSINFO structure */
   tsinfo.sc_numts = 2;
   tsinfo.sc_tsarrayp = &tslots[0];
   /* Set up DX_XPB structure */
   xpb.wFileFormat = FILE_FORMAT_VOX;
   xpb.wDataFormat = 0;
   xpb.nSamplesPerSec = 0L;
   xpb.wBitsPerSample = 0;

   /*Set up DV_TPT structure */
   dx_clrtpt (&tpt,1);
   tpt.tp_type = IO_EOT;
   tpt.tp_termno = DX_MAXDTMF;
   tpt.tp_length = 1;
   tpt.tp_flags = TF_MAXDTMF;

   /* Set up DX_IOTT structure */
   iott[0].io_fhandle = fd;
   iott[0].io_type = IO_DEV;
   iott[0].io_offset = 0;
   iott[0].io_length = MAXLEN;
   iott[0].io_offset = IO_EOT;

   /* And record from both voice channels */
   if (dx_mreciottdata(devh3, &iott[0], &tpt, &xpb, RM_TONE, &tsinfo) == -1) {
      printf("Error recording from dxxxB1C1 and dxxxB1C2\n");
      printf("error = %s\n", ATDV_ERRMSGP(devh1));
      exit(2);
   }

   /* Display termination condition value */
   printf ("The termination value = %d\n", ATDX_TERMMSK(devh1));

   /* And close three voice channels */
   if (dx_close(devh3) == -1){
      printf("Error closing devh3 \n");
      /* Perform system error processing */
      exit(3);
   }
   if (dx_close(devh2) == -1) {
      printf("Error closing devh2\n");
      /* Perform system error processing */
      exit (3);
   }
   if (dx_close(devh1) == -1) {
      printf("Error closing devh1\n");
      /* Perform system error processing */
      exit (3);
   }
   if (close(fd) == -1){
      printf("File close error \n");
      exit(1);
   }
   /* And finish */
   return;
}
```

The following example is for Windows applications.

```
#include <windows.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <srllib.h>
#include <dxxxlib.h>

#define MAXLEN 10000
```

```
main()
{
   int devh1, devh2, devh3;
   short fd;
   DV_TPT tpt;
   DX_IOTT iott[2];
   DX_XPB xpb;
   SC_TSINFO tsinfo;
   long scts;
   long tslots[32];
   char basebufp[MAXLEN];

   /* open two voice channels */
   if ((devh1 = dx_open("dxxxB1C1", NULL)) == -1) {
      printf("Could not open dxxxB1C1\n");
      exit (1);
   }

   if ((devh2 = dx_open("dxxxB1C2", NULL)) == -1) {
      printf("Could not open dxxxB1C2\n");
      exit (1);
   }

   if ((devh3 = dx_open("dxxxB1C3", NULL)) == -1) {
      printf("Could not open dxxxB1C2\n");
      exit (1);
   }

   if ((fd = dx_fileopen("file.vox", O_CREAT | O_RDWR | O_BINARY)) == -1){
      printf("File open error\n");
      exit (1);
   }

   /*
    * Get channels' external time slots
    * and fill in tslots[] array
    */

   tsinfo.sc_numts = 1;
   tsinfo.sc_tsarrayp = &scts;

   if (dx_getxmitslot (devh1, &tsinfo) == -1 )
      { /* Handle error */ }
   tslots[0] = scts;
   if (dx_getxmitslot (devh2, &tsinfo) == -1 )
      { /* Handle error */ }
   tslots[1] = scts;

   /* Set up SC_TSINFO structure */
   tsinfo.sc_numts = 2;
   tsinfo.sc_tsarrayp = &tslots[0];

   /* Set up DX_XPB structure */
   xpb.wFileFormat = FILE_FORMAT_VOX;
   xpb.wDataFormat = 0;
   xpb.nSamplesPerSec = 0L;
   xpb.wBitsPerSample = 0;

   /*Set up DV_TPT structure */
   dx_clrtpt (&tpt,1);
   tpt.tp_type = IO_EOT;
   tpt.tp_termno = DX_MAXDTMF;
   tpt.tp_length = 1;
   tpt.tp_flags = TF_MAXDTMF;
   /* Set up DX_IOTT structure */
   iott[0].io_fhandle = fd;
   iott[0].io_type = IO_DEV;
   iott[0].io_offset = 0;
   iott[0].io_length = MAXLEN;
   iott[0].io_offset = IO_EOT;

   /* And record from both voice channels */
   if (dx_mreciottdata(devh3, &iott[0], &tpt, &xpb, RM_TONE, &tsinfo) == -1) {
      printf("Error recording from dxxxB1C1 and dxxxB1C2\n");
      printf("error = %s\n", ATDV_ERRMSGP(devh1));
      exit(2);
   }

   /* Display termination condition value */
   printf ("The termination value = %d\n", ATDX_TERMMSK(devh1));

   /* And close three voice channels */
   if (dx_close(devh3) == -1){
      printf("Error closing devh3 \n");
      /* Perform system error processing */
      exit(3);
   }

   if (dx_close(devh2) == -1) {
      printf("Error closing devh2\n");
      /* Perform system error processing */
      exit (3);
   }

   if (dx_close(devh1) == -1) {
      printf("Error closing devh1\n");
      /* Perform system error processing */
      exit (3);
   }

   if (dx_fileclose(fd) == -1){
      printf("File close error \n");
      exit(1);
   }
   /* And finish */
   return;
}
```

■ See Also

- **dx_rec( )**
- **dx_play( )**
- **dx_reciottdata( )**
- **dx_playiottdata( )**

3.4.15 Voice API for Host Media Processing Programming Guide

In section *8.7.4 Streaming to Board Hints and Tips*, the default bulk queue buffer size is incorrectly described as 32 kbytes, rather than 16 kbytes. The second, fifth and sixth bullets in that section should read as follows:

- The bulk queue buffer specifies the size of the buffer used to transfer voice data between the application and HMP. This buffer is set to 16 kbytes and cannot be modified; the **dx_setchxfercnt( )** function, which is used to modify the bulk queue buffer size, is not currently supported.
- Recommendation for the high water mark: it should be based on the following:

  (size of the circular stream buffer) minus (two times the size of the bulk queue buffer)

  For example, if the circular stream buffer is 500 kbytes, and the bulk queue buffer size is 16 kbytes, set the high water mark to 468 kbytes (500-32=468).
- Recommendation for the low water mark: it should be two times the size of the bulk queue buffer size.

  Based on the previous example, since the bulk queue buffer size is 16 kbytes, set the low water mark to 32 kbytes.

The following information is added to Section 1.6, Play and Record Features:

Transaction Record

The transaction record feature allows voice activity on two channels to be summed and stored in a single file, or in a combination of files, devices, and memory. This feature is useful in call center applications where it is necessary to archive a verbal transaction or record a live conversation.

The following information is added to Chapter 8, Recording and Playback:

Transaction Record

Transaction record enables the recording of a two-party conversation by allowing two time-division multiplexing (TDM) bus time slots from a single channel to be recorded. This feature is useful for call center applications where it is necessary to archive a verbal transaction or record a live conversation. A live conversation requires two time slots on the TDM bus, but Intel voice boards today can only record one time slot at a time. No loss of channel density is realized with this feature. Voice activity on two channels can be summed and stored in a single file, or in a combination of files, devices, and/or memory.

3.5 Demonstration Software Documentation

This section contains updates to the following documents:

- Audio Conferencing API for Host Media Processing Demo Guide
- Continuous Speech Processing API for Host Media Processing Demo Guide
- IP Media Server for Host Media Processing Demo Guide
- Global Call API Demo Guide

3.5.1 Audio Conferencing API for Host Media Processing Demo Guide

The Audio Conferencing API for HMP Demo is designed to run using a series of four or fewer contiguous channels. The highest channel number in the series must be 120 or lower. Attempts to run the demo using a channel number higher than 120 will cause the demo to fail.
(IPY00009701 = PTR 34019)

In **Section 1.1, “Overview**” of **Chapter 1, “Demo Description”**, for Linux, the Audio Conferencing API for HMP Demo is located in:

```
$(INTEL_DIALOGIC_DIR)/demos/Conferencing/ConferencingDemo/Release/
```

In **Section 1.2, “Features**” of **Chapter 1, “Demo Description”**, the following Call Control feature is not demonstrated by the demo:

- RAS (Gatekeeper) support via the Global Call API
  (IPY00009238 = PTR 33768)

In **Section 2.1,”Hardware Requirements”** of **Chapter 2, “System Requirements”**, the first bullet in the **Requirements for Linux** should read:

- Machine using the Red Hat Enterprise Linux Advanced Server 3.0, Update 1 operating system with kernel version 2.6.8.1.

In **Section 4.1,”Starting the Demo”** of **Chapter 4, “Running the Demo”**, the following changes apply:

- In Step 1, for Linux, the demo is started by executing the following command:

  ```
  ./ConferencingDemo
  ```

- Step 2 of the procedure should read: “Make an IP call into the system using two IP endpoints”.

3.5.2 Continuous Speech Processing API for Host Media Processing Demo Guide

The Continuous Speech Processing API for HMP Demo is designed to run using a series of four or fewer contiguous channels. The highest channel number in the series must be 120 or lower. Attempts to run the demo using a channel number higher than 120 will cause the demo to fail.
(IPY00009701 = PTR 34019)

For Linux, the Continuous Speech Processing API for HMP Demo is located in:

```
$(INTEL_DIALOGIC_DIR)/demos/SpeechProcessing/CSPDemo/Release/
```

The demo is started by executing the following command:

```
./CSPDemo -<option>
```

3.5.3 IP Media Server for Host Media Processing Demo Guide

The IP Media Server for HMP Demo is designed to run using a series of four or fewer contiguous channels. The highest channel number in the series must be 120 or lower. Attempts to run the demo using a channel number higher than 120 will cause the demo to fail.
(IPY00009701 = PTR 34019)

The following changes and corrections apply:

- The name of this document was originally titled *IP Media Server (Global Call) Demo Guide*.
- In **Section 3.1.1, "Configuration File Location"**, of **Chapter 3, "Preparing to Run the Demo"**, the location of the file for Linux should be:

  $(INTEL_DIALOGIC_DIR)/demos/IPMediaServer/Release/

- In **Section 3.2, "Compiling and Linking"** of **Chapter 3, "Preparing to Run the Demo"**, the path given in the example for compiling the entire project should be:

  $(INTEL_DIALOGIC_DIR)/demos/IPMediaServer/

  and the path given in the example for compiling an individual module should be:

  $(INTEL_DIALOGIC_DIR)/demos/IPMediaServer/Modules/FaxModule

- In **Section 4.1, “Starting the Demo”** of **Chapter 4, “Running the Demo”**, the information in the Linux subsection should read:

  The executable is located in:

  $(INTEL_DIALOGIC_DIR)/demos/IPMediaServer/Release/

  Run the executable from this location by typing: ./IPMediaServer

- In **Section 5.1.1, “Demo Source Code Files”**, of **Chapter 5, “Demo Details”**, the path for Linux to the demo source code files in Table 3 should be:

  $(INTEL_DIALOGIC_DIR)/demos/IPMediaServer

  and the path for Linux to the demo source code files in Table 4 should be:

  $(INTEL_DIALOGIC_DIR)/demos/IPMediaServer/Modules

- In **Section 5.1.2, “PDL Files”**, of **Chapter 5, “Demo Details”**, the path for Linux to the PDL files in Table 5 should be:

  $(INTEL_DIALOGIC_DIR)/demos/Shared/pdl_Linux

3.5.4 Global Call API Demo Guide

The Global Call API Demo is designed to run using a series of four or fewer contiguous channels. The highest channel number in the series must be 120 or lower. Attempts to run the demo using a channel number higher than 120 will cause the demo to fail. (IPY00009701 = PTR 34019)

For Linux, the Global Call API Demo is located in:

```
$(INTEL_DIALOGIC_DIR)/demos/gc_basic_call_model/
```

The demo is started by executing the following command:

```
gc_basic_call_model
```