

Intel® Infrastructure DSP Solution Version 1.1

Software Release Notes

July 2007

Revision 1.0



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Intel products are not intended for use in medical, life-saving, life-sustaining, critical control or safety systems, or in nuclear-facility applications. Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The Software for IXP4XX Network Processors may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

MPEG is an international standard for video compression/decompression promoted by ISO. Implementations of MPEG CODECs, or MPEG enabled platforms may require licenses from various entities, including Intel Corporation.

This Release Notes and the software described in it are furnished under license and may only be used or copied in accordance with the terms of the license. The information in this document 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.

Except as permitted by such license, no part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the express written consent of Intel Corporation.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting Intel's website at <http://www.intel.com>.

Intel, Intel logo, and Intel XScale, are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © Intel Corporation 2007



Contents

1	Introduction	5
	1.1 Terminology	5
	1.2 Reference Documents	6
2	Overview	7
	2.1 Features	7
3	Release Components	10
	3.1 Software Components	10
	3.1.1 Makefile	10
	3.1.2 Header Files	11
	3.1.3 Plug-in Configuration Files	11
	3.1.4 Binary Files for Linux*	12
	3.1.5 Drivers and DSP Codelet	12
4	Installing and Building the Software.....	13
	4.1 Linux*	13
	4.1.1 Requirements for the Intel® Infrastructure DSP Solution	13
	4.1.2 Installing the Software.....	13
	4.1.3 Installing SRTP Source Code	15
	4.1.4 Building the Software.....	16
5	Testing the Software in Linux*.....	23
6	Demo ‘Screen Shot’ for Intel® IXP435 Multi-Service Residential Gateway Reference Platform	27
7	Limitations.....	35
	7.1 Fluctuation of CPU Utilization with “top” Command Measurement	35
	7.2 G.723.1 Latency for Multi-Frame-Per-Packet (MFPP)	35
	7.3 Improper RFC2833 tones communication between channels when SRTP is enabled	36
	7.4 Unexpected events reporting of Phase Reversal and Silence detection	36



Revision History

Revision Number	Description	Revision Date
1.0	Updated for Intel® Infrastructure DSP Solution Version 1.1 release	July 2007
001-a	Pre-production release	May 2007



1 Introduction

Intel® Infrastructure DSP Solution Version 1.1 is a software library that provides basic voice-processing functionality for Voice-over-Internet Protocol (VOIP) and residential-gateway applications for the Intel® IXP4XX Product Line of Network Processors. A demo program, supported on Linux* platform, is available to exercise the functionalities of the DSP Software library for evaluation purposes.

This document defines the features, components and installation process of the Intel Infrastructure DSP Solution Version 1.1.

In this document, all uses of “DSP software” refer to the Intel® Infrastructure DSP Solution Version 1.1 unless otherwise mentioned.

1.1 Terminology

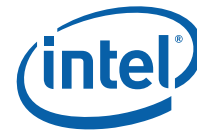
Term	Description
AGC	Automatic Gain Control
ALC	Automatic Level Control
API	Application Programmer Interface
CNG	Comfort Noise Generation
CODEC	Coder Decoder
DP	Development Platform
DPG	Development Platform Gateway
DSP	Digital Signal Processing
DTMF	Dual Tone Multi Frequency
EC	Echo Cancellation
FSK	Frequency Shift Keying
FXO	Foreign Exchange Office
FXS	Foreign Exchange Subscriber
GPL	General Public License
HSS	High Speed Serial
IP	Internet Protocol



Term	Description
LSP	Linux Support Package
NPE	Network Processor Engine
PLC	Packet Loss Concealment
RFC	Request For Comments
RTP	Real-time Transport Protocol
SLIC	Subscriber Line Interface Circuit
SRTP	Secure RTP
TDM	Time Division Multiplexing
USCI	Unified Speech Codec Interface
VAD	Voice Active Detection
VoIP	Voice Over IP

1.2 Reference Documents

Document
<i>Intel® Infrastructure DSP Solution Version 1.1 API Reference Manual</i>
<i>Intel® Infrastructure DSP Solution Version 1.1 Programmer's Guide</i>
<i>Intel® Infrastructure DSP Solution Version 1.1 Codelet Demo Guide</i>
<i>Intel® IXP400 Software Version 2.4 Software Release Notes</i>



2 Overview

The Intel® Infrastructure DSP Solution Version 1.1 provides a major enhancement by enabling customers to plug-in/add IP in the form of DSP modules making the core DSP algorithms independent of access layer software. Also, two drivers for SLIC interface and HSS have been provided and these have been targeted to work on Intel® IXP435 Multi-Service Residential Gateway Reference Platform, Intel® IXDP465 Development Platform and the Intel® IXDPG425 Network Gateway Reference Platform and these are in compliance with GPL/BSD license. Codelet has also been significantly enhanced to use socket and BSD based SRTP from CISCO*.

The Intel Infrastructure DSP Solution is an object software library. It is designed to be applied in conjunction with the Intel IXP4XX product line's base software (Intel® IXP400 Software Version 2.4).

The following summarizes the software and hardware requirements for development:

- Software
 - Intel® IXP400 Software Version 2.4
- Linux
 - Refer to Intel® IXP400 Software Version 2.4 Software Release Notes for requirements on Linux Support Package
- Platforms
 - Intel® IXP435 Multi-Service Residential Gateway Reference Platform
 - Intel® IXDP465 Development Platform
 - Intel® IXDPG425 Network Gateway Reference Platform

Example source code is provided in the “codelets” subfolder of the software release to highlight how the application layer can use the DSP functions through the API. The example code is designed to operate on platforms stated above.

2.1 Features

The DSP software provides the basic components and media-processing capability required for VoIP applications.

The major features supported in this release are API (USCI) based plug-in speech CODECs, Line Echo Canceller and separate driver modules for SLIC and HSS. The main features being supported are as below:

- Line echo cancellation up to 128ms for narrowband or up to 64ms for wideband
- G.711 μ -law and A-law CODEC with VAD and CNG support
- G.723.1 and G.729ab with VAD and CNG support
- G.726 with 16, 24, 32 and 40 Kbps rates and RFC3551 and I.366 Annex E packing formats
- G.722 wideband codec
- Dynamic switching of codec on the fly with automatic switching of decoder types according to the received RTP packets
- Packet loss concealment (PLC) for G.711, G.726 and G.722.



- Support for multiple frames per packet
- Dynamic changing of the number of frames per packet on the fly
- Automatic Gain Control (AGC) support for encoder, with provision for manual setting with mute
- Automatic Level Control (ALC) support for decoder, with provision for manual setting with mute
- DTMF generation and detection
- Modulated-tone generation capability
- Fax-tone detection (CNG, CED, and V.21-1650 Hz tone)
- Tone Disabler in NET component. Detects 2100 Hz tone with periodic phase-reversals and report events on Tx and Rx direction separately
- T30 Preamble Detector in NET component. Reports T30 preamble events on both Tx and Rx directions separately
- APIs in NET component for ToneDisabler for user programmable silence threshold level and silence duration
- APIs to re-enable reporting of events “phase-reversal in 2100 Hz tone” and “T30 Preamble” for both Tx and Rx directions
- Generation and receipt of FSK modem signals for caller ID
- Call-progress-tone generation for the United States, Japan, and China¹
- Dynamic DTMF tone clamping
- RFC-2833 tone-event support for DTMF with variable frame rate
- Dynamic/Adaptive Jitter Buffer algorithm
- Additional statistics required for extended report as per RFC3611. Statistics provided are:
 - Maximum jitter
 - Minimum jitter
 - Mean jitter
 - Standard deviation
 - Jitter buffer maximum delay
 - Jitter buffer absolute maximum delay
 - Jitter buffer nominal delay
 - Jitter discard rate
 - Echo Return Loss Enhancement (ERLE)
- Audio mixer component is enhanced to support up to four Conference Calls simultaneously, with up to three parties per conference call.
- User customizable control API
- Audio player for G.711 and G.729 recorded data
- Digital gain control in the TDM front end
- TDM switch (normal mode with echo cancellation or low latency mode)
- User-defined tones for tone generation and detection
- System configuration API
- Version query
- Support of Linux

¹ In this document, all references to China refer to the People's Republic of China.



This release does not support codec G.729.1, T.38, V.17, V.29, V.27Ter and V.21 fax modulation /demodulation and Acoustic Echo Cancellation.

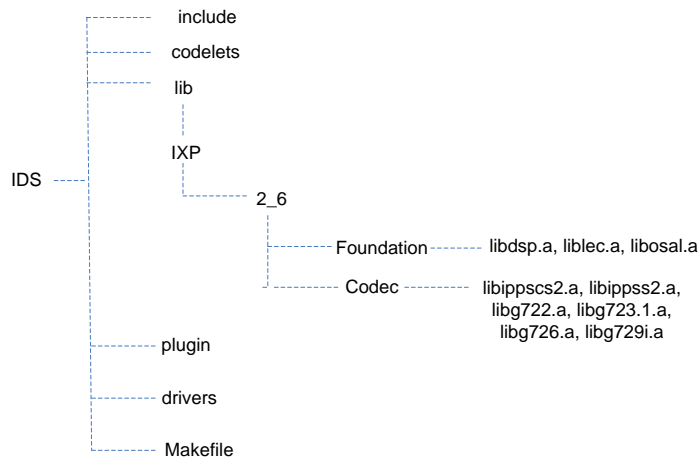


3 Release Components

The Intel® Infrastructure DSP Solution Version 1.1 software packages, Release Notes, Programmer's Guide, Codelet Demo Guide and API Reference Manual are available on the Intel Software Download Web page.

3.1 Software Components

The Intel® Infrastructure DSP Solution Version 1.1 release Package consists of the following directory structures:



Note: The IXDPG425 gateway reference platform has specific /IDS/patch directory to include the platform-specific kernel patch file (*GPL_IDS_Timesys6.1_Kernel_2.6.16.11_ixdpg425.patch*) and readme file (*Readme.txt*). All other figures shown in this document will not illustrate this platform-specific patch directory.

3.1.1 Makefile

The following files are in the top-level directory:

Component	Description
Makefile	Intel® IXP4XX Product Line Software Release top-level Makefile with updates to enable DSP-codelet related source code.



3.1.2 Header Files

The header files available in the top level include directory are as follows:

Component	Description
FWTypes.h	Contains type definitions.
msgdef.h	Contains message header definitions and function prototypes for sending control messages and receiving reply messages and event notifications.
dspcis.h	The core API. This file is the component interface specification.
dspcfg.h	Contains the API prototypes to configure and initialize the software module.
dspcfg2.h	Additional API prototypes to configure and initialize the software module.
dsptone.h	Contains the API prototypes to add new tone definitions.
ToneJP.h	Contains the definition of Japan call-progress-tone IDs.
ToneUS.h	Contains the definition of U.S. call-progress-tone IDs.
TonePRC.h	Contains the definition of China call-progress-tone IDs
SCError.h	Contains the common error code.
HssAcc.h	Contains type definitions for HSS access layer modified to support OSAL.
IxHssAcc.h	Contains type definitions for HSS access layer
IxIDS_SlicCodec.h	Contains the public APIs of the FXS/FXO devices.
usc_base.h	Defines USC base function APIs
usc_ec.h	Defines USC echo canceller APIs
usc.h	Defines USC speech codec APIs

3.1.3 Plug-in Configuration Files

The plug-in folder contains the plug-in configuration files.

Component	Description
PlugInConfig.h	This is mainly used to define all plug-in header files.
PlugInConfig.c	Configure the speech CODECs/EC plug-ins. Adds entry for either codec or EC (any order) plug-ins in the plug-in list.
PlugInDefs.h	Contains various plug-in interface types.



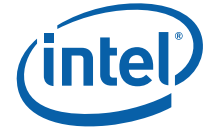
3.1.4 Binary Files for Linux*

Component	Description
libdsp.a	This is the core library, present in directory <workdir>/IDS/lib/IXP/2_6/Foundation/
liblec.a	This is Line Echo Canceller library, present in directory <workdir>/IDS/lib/IXP/2_6/Foundation/ (Note: Acoustic Echo Canceller is not supported in this version)
libippscs2.a, libippss2.a, libg722.a, libg723.1.a, libg726.a, libg729i.a	These are plug-in speech CODECs provided by Intel, present in directory <workdir>/IDS/lib/IXP/2_6/Codec

Note: <workdir> refers to the working directory or the directory in which the Intel Infrastructure DSP Solution is installed.

3.1.5 Drivers and DSP Codelet

Component	Description
SLIC	Source files for the FXS/FXO device driver
HSS	Source files for the HSS driver
dspApp	Source files for Linux*-specific DSP user-mode application



4 Installing and Building the Software

The Intel® Infrastructure DSP Solution Version 1.1 runs over Linux* operating system with Intel® IXP400 Software Version 2.4.

Prerequisite software packages:

- Refer to *Intel® IXP400 Software Version 2.4 Software Release Notes* Table 2.

4.1 Linux*

4.1.1 Requirements for the Intel® Infrastructure DSP Solution

The table below lists the files needed to build the DSP software. These requirements assume that prerequisite software packages stated in *Intel® IXP400 Software Version 2.4 Software Release Notes* Table 2 are available for installation.

Component	Description
BSD_IDS_HSS_Drv_V1_1.zip	Intel® Infrastructure DSP Solution Version 1.1 release with HSS driver, SLIC driver, Codec Library, Codelets, Foundation Library and patches
BSD_IDS_SLIC_Drv_V1_1.zip	
IPL_IDS_CodecsLibrary_V1_1.zip	
IPL_IDS_Codelets_V1_1.zip	
IPL_IDS_FoundationLibrary_V1_1.zip	

Note: The IXDPG425 platform requires Kernel Patch zip file (*GPL_IDS_LinuxLSP_ixdpg425_Patch.zip*).

4.1.2 Installing the Software

4.1.2.1 Prerequisites

Refer to *Intel® IXP400 Software Version 2.4 Software Release Notes* Section 5.1 to install the desired development platform LSP and setup host development system.

4.1.2.2 Installing Intel® IXP400 Software Version 2.4

Refer to *Intel® IXP400 Software Version 2.4 Software Release Notes* Section 5.2, from step 1 to step 11 for the installation procedures. Step 1 to step 11 of Section 5.2 installation procedures include procedures to:



- Create working directory, *<workdir>* (step 1)
- Setup LSP kernel source tree (step 2 and step 3)
- Extract Intel® IXP400 Software Version 2.4 zip files (step 4)
- Extract NPE Microcode Image zip file (step 5)
- Configure and setup environment variables (step 6, step 7, and step 8). Refer to *Intel® IXP400 Software Version 2.4 Software Release Notes* Table 5 to set the appropriate values for environment variables.
- Patch Intel® IXP400 Software Linux Ethernet Driver (step 9, step 10, and step 11)
- For the IXDPG425 platform only, unzip the kernel patch zip file (GPL_IDS_LinuxLSP_ixdpg425_Patch.zip). Copy the patch file (GPL_IDS_Timesys6.1_Kernel_2.6.16.11_ixdpg425.patch) into linux kernel directory (example: *<workdir>/linux*). Patch the kernel with the following command:

```
patch -p1 < GPL_IDS_Timesys6.1_Kernel_2.6.16.11_ixdpg425.patch
```
- For the IXDPG425 platform only, edit the Makefile *<workdir>/ixp400_xscale_sw/Makefile_ixp42X*, line no 57 to include `IX_CFLAGS += -D__ixp42X -DIXDPG425`

Make sure the clock tick rate for the system timer clock is set according to the physical hardware oscillator. Check the appropriate `CLOCK_TICK_RATE` setting defined in the `linux/include/asm-arm/arch-<ixp4xx>/timex.h` file. Clock tick rate value is defined based on the oscillator frequency multiplied by two, in unit Hz.

For example, on the Intel® IXDP465 Development Platform, the `CLOCK_TICK_RATE` setting is $33.33 \text{ MHz} * 2 = 66660000 \text{ Hz}$:

Intel® IXDP465 Development Platform(33.330 MHz oscillator)
<code>linux/include/asm-arm/arch-ixp4xx/timex.h</code> <code>#define FREQ 66660000</code>

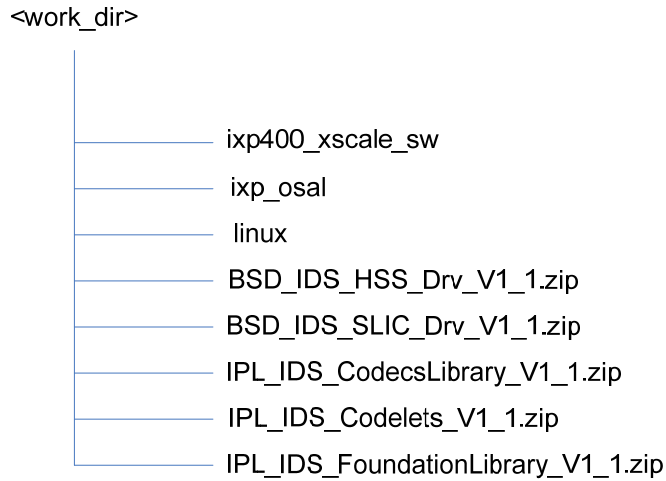
Note: For VoIP and Fax applications, 33.33MHz system reference clock and external HSS clock source should be used.

4.1.2.3 Installing Intel® Infrastructure DSP Solution Version 1.1

1. Copy the following Infrastructure DSP solution compressed files into the working directory:
 - IPL_IDS_FoundationLibrary_V1_1.zip
 - IPL_IDS_Codelets_V1_1.zip
 - IPL_IDS_CodecsLibrary_V1_1.zip
 - BSD_IDS_SLIC_Drv_V1_1.zip
 - BSD_IDS_HSS_Drv_V1_1.zip



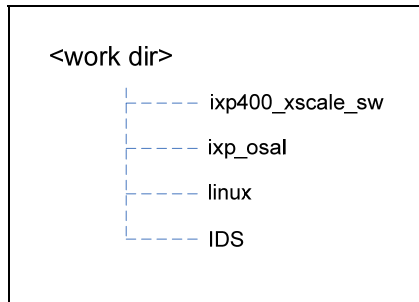
- At this point, the working directory tree has the following structure.



- Unzip the zip files by giving the following commands:

```

unzip IPL_IDS_FoundationLibrary_V1_1.zip
unzip IPL_IDS_Codelets_V1_1.zip
unzip IPL_IDS_CodecsLibrary_V1_1.zip
unzip BSD_IDS_SLIC_Drv_V1_1.zip
unzip BSD_IDS_HSS_Drv_V1_1.zip
        
```
- This extracts the .zip files and the work directory structure has the following subdirectories:



4.1.3 Installing SRTP Source Code

Download SRTP v1.4.1 source code from the following location <http://srtp.sourceforge.net/> and install in <workdir>. Unzip SRTP source code to srtp directory.

```

cd <workdir>
gunzip -c srtp-1.4.1.tar | tar -xf -
    
```

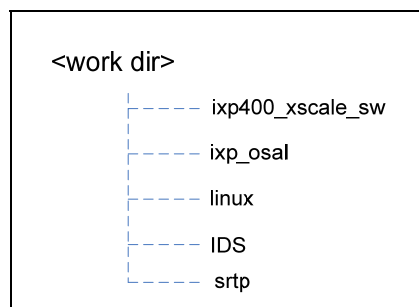


In case the SRTP source code directory is different, modify the `SRTP_BASE` environment variable value accordingly (refer to [Section 4.1.4.1](#)). Make the following changes in the Makefile as below:

```
cd <workdir>/srtp
./configure
vi Makefile
Change line no 31
CC=gcc
to
CC= /opt/timesys/toolchains/armv5b-linux/bin/armv5b-linux-gcc
```

4.1.4 Building the Software

Go to the directory `<workdir>` (`cd <workdir>`). Verify that the work directory structure has the following subdirectories:



4.1.4.1 Editing the Environment Variables for Intel® Infrastructure DSP Solution

Edit the environment file `ixp400_xscale_sw/buildUtils/environment.linux.sh` to include the following parameters for Intel® Infrastructure DSP Solution compilation. The environment variables specified in Section 4.1.2.2 of this document (which refers to *Intel® IXP400 Software Version 2.4 Software Release Notes* Section 5.2 step 6) should be maintained.

```
LINUX_SRC=<workdir>/linux
export LINUX_SRC

IX_CONTROLLED_COUNTRY_BUILD=1
export IX_CONTROLLED_COUNTRY_BUILD

Note: Include this environment variable only if you do not use the crypto version of the Access Library Software
```



<pre>MYCSR_BASE=\$IX_XSCALE_SW export MYCSR_BASE</pre>
<pre>TOOLENV=linux export TOOLENV</pre>
<pre>MYDSR_BASE=<workdir>/IDS export MYDSR_BASE</pre>
<pre>SRTP=1</pre> <p>Note: Only include this environment variable if SRTP is used, else do not define this environment variable.</p>
<pre>SRTP_BASE=<workdir>/srtp export SRTP_BASE</pre> <p>Note: Only include this environment variable if SRTP is used, else do not define and export this environment variable.</p>
<pre>IX_LINUXVER=2.6 export IX_LINUXVER</pre>

Note: As an option, single thread can be configured to handle receive functionality for all sockets instead of default setting of one socket per thread. To enable this functionality, edit <workdir>/IDS/codelets/dspApp/Makefile to include the following line:

```
CFLAGS += -DIX_DSP_CODELET_SOCKET_SINGLE_THREAD
```

Note: For Intel® IXP435 Multi-Service Residential Gateway Reference Platform, the codelet application creates four socket threads by default. Two sockets threads are assigned to the two FXS ports, one socket thread is assigned to the FX0 port, and fourth one is reserved for future use.

Note: For Intel® IXDPG425 Network Gateway Development Platform, if external HSS clock is used, the clock direction for the HSS Tx port shall be configured to Input. Edit <workdir>/IDS/codelets/dspApp/IxIDS_XScale.h line no 195 and <workdir>/IDS/drivers/HSS/include/HSSConfig.h line no 233 to change the IX_HSSACC_SYNC_CLK_DIR_OUTPUT to IX_HSSACC_SYNC_CLK_DIR_INPUT

4.1.4.2 Edit PlugInConfig.c

Extract of IDS/plugin/PlugInConfig.c is shown below:

**PlugInConfig.c**

```
#include "PlugInConfig.h" /* The global PlugInList*/
PLUG_IN_LIST_BEGIN

    /******Don't edit or modify above this line******/
    /*Add plug-in entries below */
    USCI_CODEEC_PLUG_IN (USC_G722SB_Fxns, XCODER_TYPE_G722, 0)
    USCI_CODEEC_PLUG_IN(USC_G729A_Fxns, XCODER_TYPE_G729A, 0)
    USCI_CODEEC_PLUG_IN(USC_G726_Fxns, XCODER_TYPE_G726_40, 0)
    USCI_CODEEC_PLUG_IN(USC_G726_Fxns, XCODER_TYPE_G726_32, 0)
    USCI_CODEEC_PLUG_IN(USC_G726_Fxns, XCODER_TYPE_G726_24, 0)
    USCI_CODEEC_PLUG_IN(USC_G726_Fxns, XCODER_TYPE_G726_16, 0)
    USCI_CODEEC_PLUG_IN(USC_G723_Fxns, XCODER_TYPE_G723, 0)

    /*Echo canceller plug_ins*/
        USCI_EC_PLUG_IN(USC_LECINT_Fxns, XEC_TYPE_LINE_EC, 0)
    USCI_EC_PLUG_IN(USC_ECINT_Fxns, XEC_TYPE_AEC, 0)

    /******Don't edit or modify beyond this line******/

PLUG_IN_LIST_END
```

Note: Echo Cancellor plug-in stated in PlugInConfig.c refers to Line Echo Cancellor.

Examples**Case 1**

If you need to plug-in only G726, rate-40Kbps, then select

```
USCI_CODEEC_PLUG_IN (USC_G726_Fxns, XCODER_TYPE_G726_40, 0)
```

and comment all other plug-ins between the lines PLUG_IN_LIST_BEGIN and PLUG_IN_LIST_END.

This example demonstrates the method to build G.726 codec without Line Echo Cancellor. However, in normal usage model, Line Echo Cancellor is required for VOIP application.



Case 2

If you want to select line echo canceller, then select

```
USCI_EC_PLUG_IN (USC_LECINT_Fxns, XEC_TYPE_LINE_EC, 0)
```

and comment all others between the lines PLUG_IN_LIST_BEGIN and PLUG_IN_LIST_END.

This example shows how to build Line Echo Canceller plug-in only. This example demonstrates that the application requires default G.711 u-law and G.711 a-law codecs with Line Echo Canceller. All other codecs will not be included in this case.

Case 3

If you want to select "G726, rate-40Kbps" and "line echo canceller", then select the following two lines

```
USCI_CODEC_PLUG_IN(USC_G726_Fxns, XCODER_TYPE_G726_40, 0)
```

```
USCI_EC_PLUG_IN (USC_LECINT_Fxns, XEC_TYPE_LINE_EC, 0)
```

Comment all other plug-ins from PlugInConfig.c.

4.1.4.3 Edit PlugInConfig.h

Extract of IDS/plugin/PlugInConfig.h is shown below:

PlugInConfig.h	
<pre> #ifndef __PLUGINCONFIG_H__ #define __PLUGINCONFIG_H__ #include "PlugInDefs.h" DECLARE_PLUG_IN_LIST /*Don't edit or modify above this line*/ /*Add Plug-in Header file below*/ #include "g729usc.h" #include "g723usc.h" #include "g726usc.h" #include "g722usc.h" #include "lecinterface.h" /*Line echo canceller*/ #include "ecusc_int.h" /*IPP Acoustic echo canceller*/ </pre>	



```
#include "g729_lusc.h"

/*Don't edit or modify beyond this line*/

#endif /*__PLUGINCONFIG_H__*/
```

Note: Acoustic Echo Cancellor (#include "ecusc_int.h") and G.729.1 codec (#include "g729_lusc.h") are not supported in this release. It is recommended that both the plug-ins should be commented.

Examples

Case 1

If you need to plug-in only G,726, rate-40Kbps, then select

```
#include "g726usc.h"
```

and comment all other lines between the lines DECLARE_PLUG_IN_LIST and #endif.

This example shows how to build G.726 codec without Line Echo Cancellor. However, in normal usage model, Line Echo Cancellor is required for VOIP application.

Case 2

If you want to select line echo canceller, then select

```
#include "lecinterface.h"
```

and comment all other lines between the lines DCLARE_PLUG_IN_LIST and #endif.

This example shows how to build Line Echo Cancellor plug-in only. This example demonstrates that the application requires default G.711 u-law and G.711 a-law codecs with Line Echo Cancellor. All other codecs will not be included in this case.

Case 3

If you want to select G,726, rate-40Kbps and line echo canceller, then select the following two lines:

```
#include "g726usc.h"
```

```
#include "lecinterface.h"
```

and comment all other lines between the lines DCLARE_PLUG_IN_LIST and #endif.



4.1.4.4 Building zImage and Intel® IXP400 Software

1. After completing all the procedures mentioned in previous sections, execute the following to include the environment variables modification for the DSP software:

```
cd <workdir>/ixp400_xscale_sw
source buildUtils/environment.linux.sh
```

2. To build bootable kernel image, zImage, refer to *Intel® IXP400 Software Version 2.4 Software Release Notes* Section 5.3 step 1 to step 4. (For the IXDPG425 platform, after executing `make $IX_KBUILD_CROSSCOMPILE ixp42x_defconfig` and before making zImage, use `make $IX_KBUILD_CROSSCOMPILE menuconfig` to select IXDPG425 platform by selecting System Type --> Intel Ixp4xx Implementation Options --> IXDPG425.) This will build and copy zImage into /tftpboot/ directory.
3. To build Ethernet Kernel Module, *ixp400_eth.ko*, refer to *Intel® IXP400 Software Version 2.4 Software Release Notes* Section 5.4.1 step 1 to step 4. This will build and copy *ixp400_eth.ko* to the embedded target file system on the host machine.
4. To build Intel® IXP400 Software Version 2.4, refer to *Intel® IXP400 Software Version 2.4 Software Release Notes* Section 5.4.2. This section explains the build procedure for *ixp400.ko* and *IxNpeMicrocode.dat*. Follow the procedures to copy both *ixp400.ko* and *IxNpeMicrocode.dat* into appropriate *TARGET_FILE_SYSTEM* directory.

4.1.4.5 Building SRTP Module

Build the srtp module by giving the make command:

```
cd <workdir>/srtp
make all
```

This creates the library *libsrtplib.a* in the directory `<workdir>/srtp`.

4.1.4.6 Building the Intel® Infrastructure DSP Solution

Now go to the directory named `IDS` and follow the steps as below:

```
cd <workdir>/IDS
make all
```

This creates the files `<workdir>/IDS/drivers/HSS/lib/hssdriver.ko`, `<workdir>/IDS/drivers/SLIC/lib/ixp400_codec.ko`, `<workdir>/IDS/codelets/dspApp/IxDspCodeletApp`, `<workdir>/IDS/plugin/libplug.a` and copy the files into `<workdir>/IDS/lib`



directory. Alternatively, follow the steps from 4.1.4.7 to 4.1.4.10 to generate the above files.

4.1.4.7 Building Plug-in Module

Build DSP plug-in module by giving the `make plug` command. This builds the plug-in binary to be linked with the DSP software.

```
cd <workdir>/IDS
make plug
```

This creates `<workdir>/IDS/plugin/libplug.a` and is copied into `<workdir>/IDS/lib` directory.

4.1.4.8 Building DSP Application

Build DSP application by giving the `make app` command. This builds the application "IxDspCodeletApp". This executable contains the plug-in components selected in `PlugInConfig.c` file.

```
cd <workdir>/IDS
make app
```

The executable `<workdir>/IDS/codelets/dspApp/IxDspCodeletApp` is copied into the directory `<workdir>/IDS/lib`.

4.1.4.9 Building SLIC Driver

```
cd <workdir>/IDS
make slic
```

This creates `<workdir>/IDS/drivers/SLIC/lib/ixp400_codec.ko` and is copied into the directory `<workdir>/IDS/lib`.

4.1.4.10 Building HSS Driver

```
cd <workdir>/IDS
make hssdriver
```

This creates `<workdir>/IDS/drivers/HSS/lib/hssdriver.ko` and is copied into the directory `<workdir>/IDS/lib`.



5 Testing the Software in Linux*

A number of steps must be taken at this time in order to support cross-platform development between the Linux host-development system and the target platform.

The general cross-development setup referred to in this Release Notes requires that the following services are set up and configured:

1. A DHCP server running on the host machine — To provide an IP address for the RedBoot* boot loader on the target platform and the Linux kernel.

For example, the file `/etc/dhcpd.conf` should have an entry similar to the following:

```
subnet 192.168.0.0 netmask 255.255.255.0 {
    host ixp465 {
        hardware ethernet 00:03:47:e1:a4:5e;
        fixed-address 192.168.0.50;
        option root-path "/opt/timesys/6.1/armv5b-std/rfs";
    }
}
```

2. A TFTP service running on the host machine — To allow the RedBoot boot loader to download the kernel image.

For example, the file `/etc/xinetd.d/tftp` should be similar to the following:

```
service tftp
{
    socket_type      = dgram
    protocol        = udp
    wait            = yes
    user            = root
    server          = /usr/sbin/in.tftpd
    server_args     = -s /tftpboot
    disable        = no
}
```

3. An NFS service running on the host machine with an exported NFS directory— to provide a remote file system for the target platform and a place from which to install kernel modules for supporting the default kernel configuration that mounts the file system.

For example, the file `/etc/exports` should have a line similar to:

```
/opt/timesys/linux/6.1/armv5b-std/rfs *(rw,no_root_squash)
```

4. Enable the above services with the following commands:

```
/etc/rc.d/init.d/dhcpd restart
```



```
/etc/rc.d/init.d/xinetd restart  
/etc/rc.d/init.d/nfs start  
exportfs -a
```

- 5. A serial connection using the UART 0 port on the development board and a terminal emulator (such as Minicom*) on the host platform set to 115200 baud, 8-N-1.

Note: For IXP435 reference platform, insert jumper to J8 and J9 for usage of HSS external clock.

Note: The instructions in this Release Notes assume that the host system and the development board are connected through an Ethernet cable using the Intel® 8255x-based PCI card in the target platform. For the Intel® IXDPG425 development platform, if the RedBoot* bootloader does not support downloading of kernel image through Intel® 8255x-based PCI NIC, then the NPE Ethernet port should be used.

1. For Intel® IXDP465 Development Platform, fix four-ports-voice-module to the HSS0 port on the platform. For IXP435 reference platform, and IXDPG425 development platforms, the FXS/FXO ports are available on the platform by default and external voice module is not required.
2. For the IXDP465 and IXDPG425 development platforms, connect four telephones on FXS0 to FXS3 ports. For the IXP435 platform, only two FXS ports and two telephones can be connected.
3. Connect UART0 port to the serial port of PC using a serial cable.

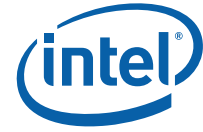
Start the target board and open a Minicom connection. Use the fconfig command to configure the board with parameters similar to those shown in the table below.

Parameter	Value
Run script at boot:	False
Use BOOTP for network configuration:	False
Local IP address:	192.168.0.50
Default server IP address:	192.168.0.100
Console baud rate:	115200
GDB connection port:	9000
Network debug at boot time:	false
Update RedBoot non-volatile configuration – are you sure (Y/N)?	Y

In the host, verify that *zImage* is located in the directory `/tftpboot` (refer to Section [4.1.4.4](#) step 2). Load and run it by doing the following:

```
Redboot> load -v -r -b % {FREEMEMLO} zImage  
Redboot> exec
```

Login as root



In the host, copy the following files to the NFS directory

/opt/timesys/linux/6.1/armv5b-std/rfs/tmp

Filename	Available in directory
ixp400.ko	<workdir>/ixp400_xscale_sw/<IX_DEVICE>/lib/linuxbe
ixp400_eth.ko	<workdir>/linux/drivers/net
hssdriver.ko	<workdir>/IDS/drivers/HSS/lib
ixp400_codec.ko	<workdir>/IDS/drivers/SLIC/lib
IxDspCodeletApp	<workdir>/IDS/codelets/dspApp
IxNpeMicrocode.dat	<workdir>/ixp400_xscale_sw/<IX_DEVICE>/lib/linuxbe

On the target platform, load the modules and run the application by executing the following commands.

Intel® IXP435 Multi-Service Residential Gateway Reference Platform	Intel® IXDP465 Development Platform or Intel® IXDPG425 Network Gateway Reference Platform
<pre>cd /tmp insmod ixp400.ko mknod /dev/ixNpe c 241 0 cat IxNpeMicrocode.dat > /dev/ixNpe insmod ixp400_eth.ko no_phy_scan=1 hss_coexist=1 mknod /dev/ixSlicModule c 252 0 mknod /dev/hssdriver c 251 0 insmod ixp400_codec.ko insmod hssdriver.ko chmod +x IxDspCodeletApp ifconfig ixp2 192.168.10.1 ifconfig ixp1 192.168.20.1 ./IxDspCodeletApp</pre>	<pre>cd /tmp insmod ixp400.ko mknod /dev/ixNpe c 241 0 cat IxNpeMicrocode.dat > /dev/ixNpe insmod ixp400_eth.ko no_phy_scan=1 mknod /dev/ixSlicModule c 252 0 mknod /dev/hssdriver c 251 0 insmod ixp400_codec.ko insmod hssdriver.ko chmod +x IxDspCodeletApp ifconfig ixp0 192.168.10.1 ifconfig ixp1 192.168.20.1 ./IxDspCodeletApp</pre> <p>Note: The parameter “no_phy_scan=1” is not used in case of Intel® IXDP465 Development Platform; therefore you can load the module ixp400_eth.ko without this parameter.</p>

Set the parameters similar to those shown in the table below:

Parameter	Value
Set country code (1:US, 81:Japan, 86:China)	1



Parameter	Value
Enter companding mode (0:A-law, 1:u-law, 2:Linear)	1
Enter slic type (0:Narrow band,1:Wide band)	0

Note: For IXP435 reference platform, companding mode should be selected as "2: Linear" when Wideband SLIC type is selected.



6 Demo 'Screen Shot' for Intel® IXP435 Multi-Service Residential Gateway Reference Platform

```
./IxDspCodeletApp
ixp400: module license 'unspecified' taints kernel.
ixp400_eth: Use default MAC address 00:02:b3:02:02:02 for port 1
ixp400_eth: Use default MAC address 00:02:b3:03:03:03 for port 2
hssdriver_init_module :: LOADED SUCCESSFULLY HSS Driver ...
ixSlic_init_module :: LOADED SUCCESSFULLY IXSLIC MODULE ...
Set country code (1:US, 81:Japan, 86:China) - 1

Existing NPEA Image ID: 0x10800201

New NPEA Image ID : 0x10900000
ixQMgrInit: IxQMgr already initialised
---- Initializing HSS Device ----
Initializing common mutex for ETH-HSS coexist
Enter companding mode (0:A-law, 1:u-law, 2:Linear) - 1
Enter slic type (0:Narrow band,1:Wide band) - 0
DLcid 21010000-01
*****
*      Intel Infrastructure DSP Solution      *
*      Release 1.1                          *
*      Jun 19 2007, 16:11:13                *
*      Intel Corporation                     *
*****
```



Processed Plug-in list

Initializing echo cancellor(s)

Channel 1:Line Echo Canceller - Done

Channel 2:Line Echo Canceller - Done

Channel 3:Line Echo Canceller - Done

Initializing decoders

Channel 1:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

Channel 2:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

Channel 3:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

Channel 4:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

Channel 5:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

Channel 6:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

Channel 7:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

Channel 8:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done



Initializing encoders

Channel 1:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

Channel 2:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

Channel 3:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

Channel 4:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

Channel 5:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

Channel 6:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

Channel 7:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

Channel 8:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

Initializing Player decoders

Channel 1:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

Channel 2:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done



Channel 3:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

Channel 4:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,

IPP_G726, IPP_G723.1 - Done

----- DSP Resource Configuration -----

Initializing SLIC

Number of TDM terminations = 3
Number of IP terminations = 8
Number of Player = 4
Number of Mixers = 4
Number of ports per Mixer = 4
Country code = 1

SLIC is initialized.

- IxDspCodelet Demo Menu -

- 0 - Print Menu
- 1 - Channel Setup
- 2 - Channel Teardown
- 3 - Show Channel Parameters
- 4 - Show Resource Parameters
- 5 - Set a Parameter
- 6 - Caller ID Demo
- 7 - Gateway and Fax Bypass Demo



- 8 - 3-Way Call Demo
- 9 - Player Demo
- 10 - SLIC APIs
- 11 - Diagnostic
- 12 - Socket Configuration
- 13 - CPU Occupancy
- 14 - CIDCW Demo
- 15 - Pulse Dial
- 16 - Set Plug-in parameter(s)
- 17 - Show Plug-in parameter(s)
- 18 - Multiconference Demo
- 19 - Exit

Please select test item -

ClientProcess started

12

```
-----  
-      IxDspCodelet Socket Menu      -  
-----
```

- 0 - Print Menus
- 1 - Channel Configuration
- 2 - Disable Channel
- 3 - Show Channel Configuration
- 4 - Show Channel Statistics
- 5 - Create Sockets
- 6 - Default Configuration
- 7 - Exit

Please select test item - 6

Enabling Default Channel Configuration



ixp0/ixp2 IPAddr of : 192.168.10.1 ixp1 IPAddr of :
192.168.20.1 is assumed

*** DSP Codelet Socket And Voice Channel Config Info ***

Chan#	RxSock#	TxSock#	Cfg	Act	Loc Port	Remote IP	Rem Port
1	10	11	Y	Y	2000	192.168.20.1	2001
2	12	13	Y	Y	2001	192.168.10.1	2000
3	14	15	Y	Y	2002	192.168.20.1	2003
4	16	17	Y	Y	2003	192.168.10.1	2002

- IxDspCodelet Socket Menu -

-
- 0 - Print Menu
 - 1 - Channel Configuration
 - 2 - Disable Channel
 - 3 - Show Channel Configuration
 - 4 - Show Channel Statistics
 - 5 - Create Sockets
 - 6 - Default Configuration
 - 7 - Exit

Please select test item - 7

- IxDspCodelet Demo Menu -



```
0 - Print Menu
1 - Channel Setup
2 - Channel Teardown
3 - Show Channel Parameters
4 - Show Resource Parameters
5 - Set a Parameter
6 - Caller ID Demo
7 - Gateway and Fax Bypass Demo
8 - 3-Way Call Demo
9 - Player Demo
10 - SLIC APIs
11 - Diagnostic
12 - Socket Configuration
13 - CPU Occupancy
14 - CIDCW Demo
15 - Pulse Dial
16 - Set Plug-in parameter(s)
17 - Show Plug-in parameter(s)
18 - Multiconference Demo
19 - Exit

Please select test item - 1

Enter the channel number (1,2,...; 0 for all channels) - 0

Coder types :

0:PassThru (for debug only)
1:G711_10ms u-Law
2:G711_10ms A-Law
3:G729a/b
4:G723.1
5:G722
```



6:G726 40Kbps

7:G726 32Kbps

8:G726 24Kbps

9:G726 16Kbps

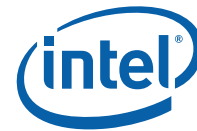
10:G729.1

Enter the coder type - 1

```
<<IxDspCodeletMsgSetupAck received (ch=1, numDspReply=10, numErr=0)>>
```

```
<<IxDspCodeletMsgSetupAck received (ch=2, numDspReply=10, numErr=0)>>
```

```
<<IxDspCodeletMsgSetupAck received (ch=3, numDspReply=10, numErr=0)>>
```



7 Limitations

7.1 Fluctuation of CPU Utilization with “top” Command Measurement

Reference Number: IXA00153759

Intel® IXP400 Software Version(s): 2.4

Platform(s): IXDP465, IXDPG425 and IXP435 platforms

Description: The observation is “top” command measurement shows fluctuation of CPU utilization between 0% and 99%.

Implication: “Top” is not suitable for measuring timer interrupt synced task, “top” measurement may show fluctuation of CPU utilization. However, voice quality measurement shows that voice quality is not impacted.

Resolution/Workaround: Please contact your Intel representative for additional information.

7.2 G.723.1 Latency for Multi-Frame-Per-Packet (MFPP)

Reference Number: IXA00177311

Intel® IXP400 Software Version(s): 2.4

Platform(s): Intel® IXDP465, Intel® IXDPG425 and Intel® IXP435

Description: While using G.723.1 codec, round-trip delay of up to 970 ms is observed. Test performed with default codelets loopback setup without any IP network delay. Voice quality measurement results are not degraded by the delay. Round-trip delay (latency) in loop-back mode without MFPP is constant at 300 ms for G.723.1, which is expected.

Implication: There is no issue while using Single-Frame-Per-Packet (SFPP); latency can be high while using Multiple-Frame-Per-Packet (MFPP).

Resolution/Workaround: Please contact your Intel representative for resolution or additional information.



7.3 Improper RFC2833 tones communication between channels when SRTP is enabled

Reference Number: IXA00165500

Intel® IXP400 Software Version(s): 2.4

Platform(s): Intel® IXDP465, Intel® IXDPG425 and Intel® IXP435

Description: After channel setup with SRTP v1.4.1 and RFC2833 enabled, if the caller repeatedly presses the DTMF push buttons, receiving terminal will not hear the DTMF tones after certain duration.

Implication: When SRTP v1.4.1 is enabled, the RFC2833 tone communication between the channels might not work properly.

Resolution/Workaround: Disable RFC2833 Event (XPARMID_TD_RFC2833E_ENABLE) and Tone Clamping (XPARMID_TD_TC) of Tone Detector while using SRTP.

7.4 Unexpected events reporting of Phase Reversal and Silence detection

Reference Number: IXA00165503

Intel® IXP400 Software Version(s): 2.4

Platform(s): Intel® IXDP465, Intel® IXDPG425 and Intel® IXP435

Description: Unexpected Phase Reversal and Silence detection events are reported, when the Phase Reversal and Silence detection events are enabled during the Fax calls.

Implication: Enabling Phase reversal and Silence detection may wrongly enable/disable echo canceller during Fax transmission.

Resolution/Workaround: No workaround; please contact your Intel representative for additional information.