Intel® Infrastructure DSP Solution Version 1.2

Software Release Notes

December 2007

Revision 1.0

Contents

Revision History

Revision Number	Description	Revision Date
1.0	Updated for Intel® Infrastructure DSP Solution Version 1.2 release	December 2007

1   Introduction

The Intel® Infrastructure DSP Solution Version 1.2 is a software library that provides basic voice-processing functionality for Voice-over-Internet Protocol (VoIP) and residential-gateway applications for the Intel® IXP46X Product Line of Network Processors, and Intel® IXP42X 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 1.2.

In this document, all uses of "DSP software" refer to the Intel® Infrastructure DSP Solution Version 1.2 unless otherwise mentioned.

1.1   Terminology

Term	Description
AGC	Automatic Gain Control
ALC	Automatic Level Control
API	Application Programming Interface
CNG	Comfort Noise Generation
CODEC	Coder Decoder
DP	Development Platform
DPG	Development Platform Gateway
DsCP	Differentiated service Code Point
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

Term	Description
HSS	High Speed Serial
iLBC	internet Low Bit rate Codec
IP	Internet Protocol
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
Intel® IXP400 Software Version 2.4 Software Release Notes
Intel® Infrastructure DSP Solution Version 1.2 Codelet Demo Guide
Intel® Infrastructure DSP Solution Version 1.2 API Reference Manual
Intel® Infrastructure DSP Solution Version 1.2 Programmer's Guide

2   Overview

The Intel® Infrastructure DSP Solution Version 1.2 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 drivers are targeted to Intel® IXDP465 Development Platform, and Intel® IXDPG425 Network Gateway Reference Platform and they 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® 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 128 ms for narrowband or up to 64 ms 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

- G.729.1 wideband and narrowband codec with bit rate negotiation

- iLBC codec with 20 ms and 30 ms frame size

- T.38 Fax as pluggable module with Illico® based data pump V.17, V.29, V.27Ter and V.21 fax modulation /demodulation

- 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 features:
  - Detects and reports periodic phase-reversals event in 2100 Hz tone and subsequent silence events on Tx, and Rx direction separately. Silence threshold level and silence duration are user programmable through APIs.
  - T30 Preamble Detector in NET component. Reports T30 preamble events on both Tx, and Rx directions separately
- Generation and receipt of FSK modem signals for caller ID
- Call-progress-tone generation for the United States, Japan, and China[1]
- 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 depending on configuration and target platform.
- Audio player: Ability to playback G.711 and G.729 encoded speech 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

This release does not support Acoustic Echo Cancellation.

---

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

3   Release Components

The Intel® Infrastructure DSP Solution Version 1.2 release consists of software packages that contains software components and release documents such as Release Notes, Programmer's Guide, Codelet Demo Guide, and API Reference Manual. All customer documents are available on request.

3.1   Software Components

The Intel® Infrastructure DSP Solution Version 1.2 release Package consists of the following directory structures.

```
IDS
 ├──── codelets
 ├──── Makefile
 ├──── drivers
 ├──── include
 ├──── plugin
 └──── lib
        └──── IXP
               └──── 2_6
                      ├──── Codec
                      │      ├──── libg722.a
                      │      ├──── libg726.a
                      │      ├──── libg723.1.a
                      │      ├──── libg729.a
                      │      ├──── libg729.1.a
                      │      ├──── libillico.a
                      │      ├──── libt38.a
                      │      ├──── libippscs2.a
                      │      ├──── libippss2.a
                      │      └──── libilbc.a
                      └──── Foundation
                             ├──── libdsp.a
                             ├──── libosal.a
                             └──── liblec.a
```
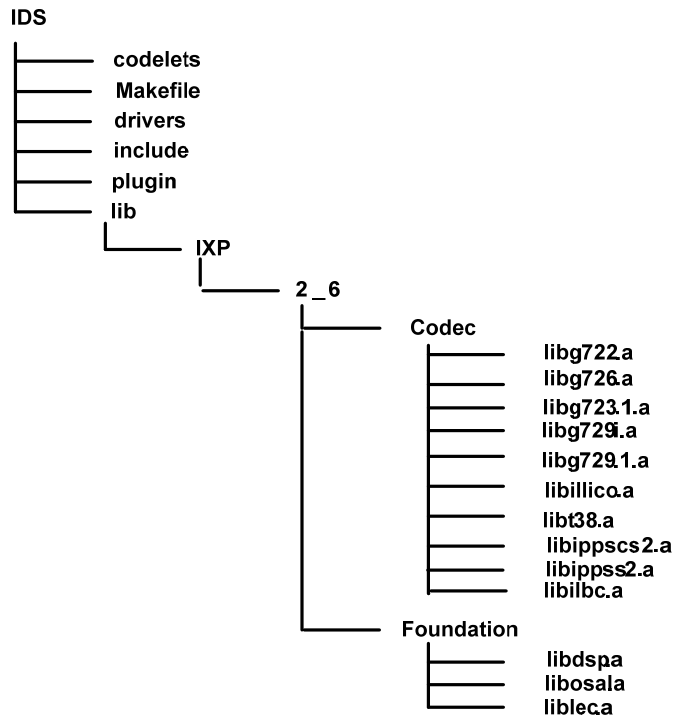
> *Note:* The Intel® IXDPG425 Network Gateway Reference Platform has specific /IDS/patch directory to include platform-specific kernel patch file (*GPL_IDS_Timesys6.1_Kernel_2.6.16.11_ixdpg425.patch*) and readme file (*Readme.txt*). Other figures in this document will not illustrate this platform-specific patch directory.

3.1.1   Make file

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
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
usc_t38.h	Defines USC T.38 fax 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 for kernel 2.6 `<workdir>/IDS/lib/IXP/2_6/Foundation/`
libosal.a	Operating Systems Abstraction Layer Library for User Space, present in directory for kernel 2.6 `<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, libg729.1.a,libilbc.a	These are plug-in speech CODECs provided by Intel, present in directory for kernel 2.6 `<workdir>/IDS/lib/IXP/2_6/Codec`
libt38.a, libillico.a	These are T38 fax plug-in component provided by Intel, present in directory for kernel 2.6 `<workdir>/IDS/lib/IXP/2_6/Codec`

Note: `<workdir>` refers to the working directory or the directory in which the Intel Infrastructure DSP Solution has been 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 1.2 runs on Linux* operating system with Intel® IXP400 Software Version 2.4.

Prerequisite software packages:

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

4.1　Linux*

4.1.1　Requirements for Infrastructure DSP Solution

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

Component	Description
BSD_IDS_HSS_Drv_V1_2.zip BSD_IDS_SLIC_Drv_V1_2.zip IPL_IDS_CodecsLibrary_V1_2.zip IPL_IDS_Codelets_V1_2.zip IPL_IDS_FoundationLibrary_V1_2.zip	Intel® Infrastructure DSP Solution Version 1.2 release with HSS driver, SLIC driver, Codec Library, Codelets, Foundation Library and patches.

Note: The IXDPG425 Network Gateway Reference 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　Intel® IXP400 Software Version 2.4

Refer to steps 1 to 11 in Section 5.2 of the Intel® IXP400 Software Version 2.4 Software Release Notes for installation procedures. The installation procedures mentioned in steps 1 to 11 in Section 5.2 of the Intel® IXP400 Software Version 2.4 Software Release Notes include the following:

- Created working directory, *<workdir>* (step 1)

- Setup LSP kernel source tree (step 2 and step 3)

- Extracted Intel® IXP400 Software Version 2.4 zip files (step 4)

- Extracted NPE Microcode Image zip file (step)

- Configured and setup environment variables (step 6, step 7 and step 8). Refer to Intel® IXP400 Software Version 2.4 Software Release Notes Table 5 Makefile Macros to set the appropriate values for environment variables.

- Obtain the most recently available Intel® IXP400 Software Linux* Ethernet Driver, Extract it and apply the patch for Intel® IXP400 Software Linux Ethernet Driver (step 9, step 10, and step 11)

- Perform the following for the IXDPG425 Network Gateway Reference 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
    ```

  - Edit the Makefile `<workdir>/ixp400_xscale_sw/Makefile_ixp42X`, line no 57 to include `IX_CFLAGS += -D__ixp42X -DIXDPG425`

Ensure that 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. On the Intel® IXDP465 Development Platform, the `CLOCK_TICK_RATE` settings are as shown in the following table:

| Intel® IXDP465 Development Platform

(33.330 MHz OSC) |
|---|
| `linux/include/asm-arm/arch-ixp4xx/timex.h` |
| `#define FREQ 66660000` |

4.1.2.3   Intel® Infrastructure DSP Solution Version 1.2

1. Copy the following Infrastructure DSP solution compressed files into the working directory:

  - IPL_IDS_FoundationLibrary_V1_2.zip

  - IPL_IDS_Codelets_V1_2.zip

  - IPL_IDS_CodecsLibrary_V1_2.zip

- BSD_IDS_SLIC_Drv_V1_2.zip

- BSD_IDS_HSS_Drv_V1_2.zip

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

\<work_dir\>
```
├─── BSD_IDS_HSS_Drv_V1_2.zip
├─── BSD_IDS_SLIC_Drv_V1_2.zip
├─── IPL_IDS_CodecsLibrary_V1_2.zip
├─── IPL_IDS_Codelets_V1_2.zip
├─── IPL_IDS_FoundationLibrary_V1_2.zip
├─── ixp400_xscale_sw
├─── ixp_osal
└─── linux
```
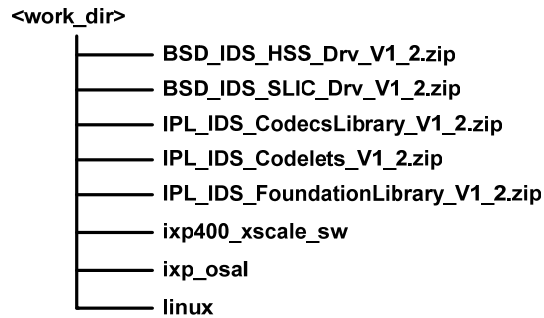
3. Unzip the zip files by giving the following commands:

unzip   BSD_IDS_HSS_Drv_V1_2.zip

unzip   BSD_IDS_SLIC_Drv_V1_2.zip

unzip   IPL_IDS_CodecsLibrary_V1_2.zip

unzip   IPL_IDS_Codelets_V1_2.zip

unzip   IPL_IDS_FoundationLibrary_V1_2.zip

4. This extracts the .zip files and the work directory structure has the following subdirectories:

\<work_dir\>
```
├─── IDS
├─── ixp400_xscale_sw
├─── ixp_osal
└─── linux
```
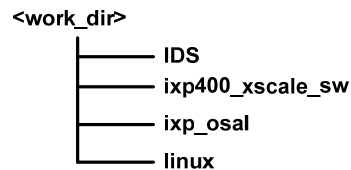
4.1.3   Installing SRTP Source Code

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:

```
<work_dir>
        ├──── IDS
        ├──── ixp400_xscale_sw
        ├──── ixp_osal
        ├──── linux
        └──── srtp
```
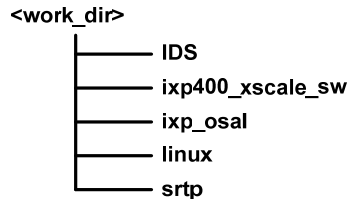
4.1.4.1   Edit Environment Variables for Intel® Infrastructure DSP Solution

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**: Only include this environment variable if you do not use the crypto version of the Access Library Software
MYCSR_BASE=$IX_XSCALE_SW export MYCSR_BASE
TOOLENV=linux export TOOLENV

MYDSR_BASE=<**workdir**>/IDS export MYDSR_BASE
SRTP=1 **Note**: Only include this environment variable if SRTP is used, else do not define this environment variable.
SRTP_BASE = <**workdir**>/srtp export SRTP_BASE **Note**: Only include this environment variable if SRTP is used, else do not define and export this environment variable.
IX_LINUXVER=2.6 Export IX_LINUXVER
IX_XSCALE_SW_VER=24 export IX_XSCALE_SW_VER

Note: For Intel® IXDPG425 Network Gateway Development Platform, if you use an external HSS clock, the clock direction for the HSS Tx port shall be configured to Input. Edit `<workdir>/IDS/codelets/dspApp/IxIDS_XScale.h` line no 199 and `<workdir>/IDS/drivers/HSS/include/HSSConfig.h` line no 241 to change '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 in the following page:

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 */
/*Codec plug-ins*/

USCI_CODEC_PLUG_IN(USC_G722SB_Fxns,XCODER_TYPE_G722,0)

USCI_CODEC_PLUG_IN(USC_G729A_Fxns,XCODER_TYPE_G729A,0)

USCI_CODEC_PLUG_IN(USC_G726_Fxns,XCODER_TYPE_G726_40,0)

USCI_CODEC_PLUG_IN(USC_G726_Fxns,XCODER_TYPE_G726_32,0)

USCI_CODEC_PLUG_IN(USC_G726_Fxns,XCODER_TYPE_G726_24,0)

USCI_CODEC_PLUG_IN(USC_G726_Fxns,XCODER_TYPE_G726_16,0)

USCI_CODEC_PLUG_IN(USC_G723_Fxns,XCODER_TYPE_G723,0)
        /*T38 Fax plug-in*/
USCI_FAX_PLUG_IN(USC_T38INT_Fxns,XFAX_TYPE_T38,0)

USCI_CODEC_PLUG_IN(USC_G7291_Fxns,XCODER_TYPE_G729_1,0)

USCI_CODEC_PLUG_IN(USC_ILBC_Fxns,XCODER_TYPE_ILBC_30MS,0)

USCI_CODEC_PLUG_IN(USC_ILBC_Fxns,XCODER_TYPE_ILBC_20MS,0)
        /*Echo cancellor plug_ins*/
USCI_EC_PLUG_IN(USC_LECINT_Fxns,XEC_TYPE_LINE_EC,0)
/********Don't edit or modify beyond this line********/

PLUG_IN_LIST_END
```

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

Examples

Case 1

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

    USCI_CODEC_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 Canceller. However, in normal usage model, Line Echo Canceller 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
| ```
#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 cancellor*/
#include "ecusc_int.h"      /*IPP Accoustic echo cancellor*/
#include "g729_1usc.h"
#include "iLBCusc.h"
#include "t38interface.h"

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

        #endif /*__PLUGINCONFIG_H__*/
``` |

> *Note:* Acoustic Echo Canceller (`#include "ecusc_int.h"`) is not supported in this release. It is recommended that the Acoustic Echo Canceller plug-in should be commented.

Examples

Case 1

If you need to plug-in only G.726, rate-40 Kbps, 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 Canceller. However, in normal usage model, Line Echo Canceller 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 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 G.726, rate-40 Kbps 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 Intel® Infrastructure DSP Solution:

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

2. To build bootable kernel image, *zImage*, refer to steps 1 to 4 in Section 5.3 of the Intel® IXP400 Software Version 2.4 Software Release Notes. For the IXDPG425 platform, after executing make $IX_KBUILD_CROSSCOMPILE ixp42x_defconfig and before making zImage, use make $IX_KBUILD_CROSSCOMPILE menuconfig to select the IXDPG425 gateway reference 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 steps 1 to 4 in Section 5.4 of the Intel® IXP400 Software Version 2.4 Software Release Notes. 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 Section 5.4.2 of the Intel® IXP400 Software Version 2.4 Software Release Notes. 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 libsrtp.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 Section 4.1.4.7 to Section 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 link with DSP Application.

```
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 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`.

4.1.4.9　　　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`.

4.1.4.10　　　Building DSP Application

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

```
cd <workdir>/IDS

make app
```

The executable `<workdir>/IDS/codelets/dspApp/IxDspCodeletApp` 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: The instructions in this Release Notes assume that the host system and Intel® IXDP465 Development Platform is connected via a cross-over Ethernet cable using the Intel® 8255x-based PCI card in the target platform. For the Intel® IXDPG425 reference platform, if the RedBoot* boot loader does not support downloading of kernel image through Intel® 8255x-based PCI NIC, then the NPE Ethernet port should be used.

For the *Intel® IXDP465 Development Platform*:
1. Fix four-ports-voice-module to HSS0 port of the Intel® IXDP465 Development Platform. For the Intel® IXDPG425 Network Gateway Reference Platform, the FXS/FXO ports are available on the platform by default and an external voice module is not required..
2. Connect four telephones to FXS0 to FXS3 ports of voice module on the Intel® IXDP465 and IXDPG425 Platforms.
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 **ifconfig** 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 of this document). 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	For Intel® IXDP465 Development Platform:
	`<workdir>/ixp400_xscale_sw/ixp46X/lib/linuxbe`
	For Intel® IXDPG425 Network Gateway Reference Platform:
	`<workdir>/ixp400_xscale_sw/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	For Intel® IXDP465 Development Platform:
	`<workdir>/ixp400_xscale_sw/ixp46X/lib/linuxbe`
	For Intel® IXDPG425 Network Gateway Reference Platform:
	`<workdir>/ixp400_xscale_sw/lib/linuxbe`

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

Intel® IXDP465 Development Platform	Intel® IXDPG425 Network Gateway Reference Platform
``cd /tmp
insmod ixp400.ko
mknod /dev/ixNpe c 241 0
cat IxNpeMicrocode.dat >
/dev/ixNpe
insmod ixp400_eth.ko
insmod ixp400_codec.ko
insmod hssdriver.ko
mknod /dev/ixSlicModule c 252 0
mknod /dev/hssdriver c 251 0
chmod +x IxDspCodeletApp
ifconfig ixp0 192.168.10.1
ifconfig ixp1 192.168.20.1

./IxDspCodeletApp``` | ```cd /tmp
insmod ixp400.ko
mknod /dev/ixNpe c 241 0
cat IxNpeMicrocode.dat >
/dev/ixNpe
insmod ixp400_eth.ko
no_phy_scan=1
insmod ixp400_codec.ko
insmod hssdriver.ko
mknod /dev/ixSlicModule c 252 0
mknod /dev/hssdriver c 251 0
chmod +x IxDspCodeletApp
ifconfig ixp0 192.168.10.1
ifconfig ixp1 192.168.20.1

./IxDspCodeletApp``` |
| **Note**: If kernel module version mismatch is seen while doing insmod, specify "-f" option.

For example, insmod –f ixp400_codec.ko | |

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:Narrowband, 1:Wideband)	0

6 Demo Screen Captures for the Intel® IXDP465 Development Platform

```
      Set country code (1:US, 81:Japan, 86:China) - 1
      Enter companding mode (0:A-law, 1:u-law, 2:Linear) -1
      Enter slic type (0:Narrow band,1:Wide band) - 0

  Existing NPEA Image ID: 0xD0000013

  New NPEA Image ID: 0x00160000
---- Initializing HSS Device ----
DLcid 01020000-01
******************************************
*      Intel Infrastructure DSP Solution    *
*      Release 1.2                          *
*      Nov 17 2007, 23:00:40                *
*      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
Channel 4:Line Echo Canceller  - Done
Initializing decoders
Channel 1:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,
 IPP_G726, IPP_G7291, IPP_iLBC, IPP_G723.1, IPP_iLBC - Done
Channel 2:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,
 IPP_G726, IPP_G7291, IPP_iLBC, IPP_G723.1, IPP_iLBC - Done
Channel 3:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,
 IPP_G726, IPP_G7291, IPP_iLBC, IPP_G723.1, IPP_iLBC - Done
Channel 4:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,
 IPP_G726, IPP_G7291, IPP_iLBC, IPP_G723.1, IPP_iLBC - Done
Channel 5:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,
 IPP_G726, IPP_G7291, IPP_iLBC, IPP_G723.1, IPP_iLBC - Done
Channel 6:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,
 IPP_G726, IPP_G7291, IPP_iLBC, IPP_G723.1, IPP_iLBC - Done
Channel 7:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,
 IPP_G726, IPP_G7291, IPP_iLBC, IPP_G723.1, IPP_iLBC - Done
Channel 8:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
```

```
        IPP_G726, IPP_G726,
        IPP_G726, IPP_G7291, IPP_iLBC, IPP_G723.1, IPP_iLBC - Done
Initializing Player decoders
Channel 1:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,
        IPP_G726, IPP_G7291, IPP_iLBC, IPP_G723.1, IPP_iLBC - Done
Channel 2:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,
        IPP_G726, IPP_G7291, IPP_iLBC, IPP_G723.1, IPP_iLBC - Done
Channel 3:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,
        IPP_G726, IPP_G7291, IPP_iLBC, IPP_G723.1, IPP_iLBC - Done
Channel 4:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,
        IPP_G726, IPP_G7291, IPP_iLBC, IPP_G723.1, IPP_iLBC - Done
Channel 5:PTM, G711A, G711Mu, IPP_G722, IPP_G729A, IPP_G726,
IPP_G726, IPP_G726,
        IPP_G726, IPP_G7291, IPP_iLBC, IPP_G723.1, IPP_iLBC - Done

Channel 1: T38 Fax Modem    - Done
Channel 2: T38 Fax Modem    - Done
Channel 3: T38 Fax Modem    - Done
Channel 4: T38 Fax Modem    - Done
-------- DSP Resource Configuration -------
 Initializing SLIC .................

   Number of TDM terminations = 4
   Number of IP terminations  = 8
   Number of Player           = 5
   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 IPAddrs of : 192.168.10.1 ixp1 IPAddrs 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 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 - 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
    Select combinations from the following options
       1:Channel (1<->2)
       2:Channel (1<->3)
       3:Channel (1<->4)
       4:Channel (2<->3)
       5:Channel (2<->4)
       6:Channel (3<->4)
       7:All Channels(1<->2,3<->4)
       8:All Channels(1<->3,2<->4)
    Enter your option - 7
    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
      11:iLBC 30ms
      12:iLBC 20ms
    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)>>
    <<IxDspCodeletMsgSetupAck received (ch=4, numDspReply=10,
    numErr=0)>>
```

7　Known Bugs/Open Issues

7.1　T38 Fax Plug-in Module Issues

Reference Number: IXA00178271

Intel® IXP400 Software Version(s): 2.4

Platform(s): Intel® IXDP465 Development Platform, and Intel® IXDPG425 Network Gateway Reference Platform

Description: The T.38 Fax plug-in module is tested using ChannelTrap (Quality logics® FaxLab) version 4.5. The following nominal profiles might fail intermittently. The results may vary between the tests when it is run multiple times, as the Fax is Analog in nature and also ChannelTrap generates anomalies.

- Orig: TX 3 Pg Non-ECM V.27ter 4800 Fastest Stats　　!Nominal Synthesized Profile
  Ans: RX 3 Pg Non-ECM MH Best Mod 200x200　　!Nominal Synthesized Profile

- Orig: TX 3 Pg ECM V.27ter 2400 Slowest Stats　!Nominal Synthesized Profile
  Ans: RX 3 Pg ECM MH Best Mod 200x200　!Nominal Synthesized Profile Failed

- Orig: TX 3 Pg Non-ECM V.29 7200 Random Stats　!Nominal Synthesized Profile
  Ans: RX 3 Pg Non-ECM MH Best Mod 300x300　　!Nominal Synthesized Profile

Implication: The Fax session may fail with modulation V.27ter 2.4k, 4.8k and V.29 7200

Resolution/Workaround: No workaround or resolution. The Intel® Infrastructure DSP Solution version 1.2 does not claim compliancy with ITU-T T.30 specifications.

7.2　Fluctuation of CPU Utilization with "top" Command Measurement Reference

Reference Number: IXA00153759

Intel® IXP400 Software Version(s): 2.4

Platform(s): Intel® IXDP465 Development Platform and Intel® IXDPG425 Network Gateway Reference Platform

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.

7.3 Latency for Multi-Frame-Per-Packet (MFPP)

Reference Number: IXA00177311/IXA00179657

Intel® IXP400 Software Version(s): 2.4

Platform(s): Intel® IXDP465 Development Platform, and Intel® IXDPG425 Network Gateway Reference Platform.

Description: Round-trip delay increase of up to 970 ms observed during the middle of the call duration for all 20 ms/30 ms codecs with multi frames per packet. When test is performed without any IP network delay by configuring the default loopback setup in the codelet, result shows no degradation in voice quality. For single frame per packet increase in the delay issue is not observed.

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: No workaround or resolution. Multi-Frame-Per-Packet is not supported for 20 ms or 30 ms codec.

7.4 Improper RFC2833 Tones Communication between Channels When SRTP is Enabled

Reference Number: IXA00165500

Intel® IXP400 Software Version(s): 2.4

Platform(s): Intel® IXDP465 Development Platform, and Intel® IXDPG425 Network Gateway Reference Platform

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.

7.5   System Hangs When G729.1 for Four channels is Selected

Reference Number: IXA00178553

Intel® IXP400 Software Version(s): 2.4

Platform(s): Intel® IXDP465 Development Platform, and Intel® IXDPG425 Network Gateway Reference Platform

Description: When you select codec G729.1 for four channels, the system hangs. This is a known problem because G729.1 codec consumes more CPU cycles; for two channels it requires approximately 57% of CPU cycles, hence if you select four channels then the system may end up with undesired behavior/state.

Implication: There is no issue while using for two channels.

Resolution/Workaround: Use g729.1 codec for only two channels

7.6   Low PSQM Score with G729.1

Reference Number: IXA00178456

Intel® IXP400 Software Version(s): 2.4

Platform(s): Intel® IXDP465 Development Platform, and Intel® IXDPG425 Network Gateway Reference Platform

Description: When you select codec G729.1 with 8 Kbps bit rate, low PSQM is observed.

Implication: There is no issue while using other bit rates.

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

Note: G.729.1 8 Kbps is not working and it is not recommended to use the same. Using 8 Kbps can result in unpredictable state of the Intel® Infrastructure DSP Solution (IDS).

7.7   Wideband CODECs Not Tested

Reference Number: Not Applicable.

Intel® IXP400 Software Version(s): 2.4

Platform(s): Intel® IXDP465 Development Platform, and Intel® IXDPG425 Network Gateway Reference Platform

Description: Wideband codec G.722 and Wideband bit rates for G.729.1 have not been tested on Intel® IXDP465 Development Platform, and Intel® IXDPG425 Network

Gateway Reference Platform due to hardware limitation. These two platforms do not have Wideband SLIC. The functionality of these Wideband codec has been tested on the Intel® IXP435 Multi-Service Residential Gateway Reference Platform.

Implication: There is no issue while using narrowband CODECs.

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

7.8 Low Voice Quality in Packet Loss Testing

Reference Number: IXA00179660

Intel® IXP400 Software Version(s): 2.4

Platform(s): Intel® IXDP465 Development Platform, and Intel® IXDPG425 Network Gateway Reference Platform

Description: In 5% IP Packet loss test environment, whenever there are two consecutive packet losses, voice quality is low and voice quality recovers in few seconds. This limitation is observed with G.729.1, G.723.1, and iLBC codecs only.

Implication: G.729.1, G.723.1, and iLBC codecs have low voice quality with 5% or more IP Packet loss.

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

7.9 T.38 Interoperability Test with Third Party T.38 Gateway

Reference Number: IXA00165848

Intel® IXP400 Software Version(s): 2.4

Platform(s): Intel® IXDP465 Development Platform

Description: The following results were observed during T.38 Interoperability Test:
- Most of the T.38 Fax Interoperability test passed.
- The T.38 Fax transmission from Infrastructure DSP Solution Gateway to third party T.38 gateway may fall back to a lower baud rate.
- The T.38 Fax transmission may fail intermittently.

Implication: The T.38 Fax transmission may fall back to a lower baud rate or T.38 Fax transmission may fail with the third party T.38 gateway.

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