Intel® Infrastructure DSP Solution Pre-Production Version 1.2

Software Release Notes

August 2007

Revision 0.2 (Engineering Draft)
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 ..................................................................................... 11
    3.1.1 Makefile ................................................................................................ 11
    3.1.2 Header Files ........................................................................................... 12
    3.1.3 Plug-in Configuration Files .................................................................... 12
    3.1.4 Binary Files for Linux* .......................................................................... 13
    3.1.5 Drivers and DSP Codelet ....................................................................... 13
4 Installing and Building the Software ..................................................................... 14
  4.1 Linux* ............................................................................................................. 14
    4.1.1 Requirements for Infrastructure DSP Solution ..................................... 14
    4.1.2 Installing the Software .......................................................................... 14
    4.1.3 Installing SRTP Source Code ................................................................. 16
    4.1.4 Building the Software .......................................................................... 17
5 Testing the Software in Linux* ............................................................................. 24
6 Demo ‘Screen Shot’ .............................................................................................. 28
7 Known Bugs/Open Issues ..................................................................................... 32
  7.1 T38 Fax plug-in module issues ......................................................................... 32
  7.2 Improper Codelet SLIC menu ID’s ................................................................. 32
  7.3 Fluctuation of CPU Utilization with “top” Command Measurement Reference .. 33
  7.4 G.723.1 Latency for Multi-Frame-Per-Packet (MFPP) ...................................... 33
  7.5 Improper RFC2833 tones communication between channels when SRTP is enabled ................................................................. 34
  7.6 Unexpected events reporting of Phase Reversal and Silence detection .......... 34
# Revision History

<table>
<thead>
<tr>
<th>Revision Number</th>
<th>Description</th>
<th>Revision Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>Initial release.</td>
<td>July 2007</td>
</tr>
<tr>
<td>002</td>
<td>updated for Pre-Production Release</td>
<td>Aug 2007</td>
</tr>
</tbody>
</table>
1 Introduction

Intel® Infrastructure DSP Solution Version 1.2 Pre-Production Release is a software library that provides basic voice-processing functionality for Voice-over-Internet Protocol (VOIP) and residential-gateway applications for the IXP46X 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 Pre-Production Release.

1.1 Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGC</td>
<td>Automatic Gain Control</td>
</tr>
<tr>
<td>ALC</td>
<td>Automatic Level Control</td>
</tr>
<tr>
<td>API</td>
<td>Application Programmer Interface</td>
</tr>
<tr>
<td>CNG</td>
<td>Comfort Noise Generation</td>
</tr>
<tr>
<td>CODEC</td>
<td>Coder Decoder</td>
</tr>
<tr>
<td>DP</td>
<td>Development Platform</td>
</tr>
<tr>
<td>DPG</td>
<td>Development Platform Gateway</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processing</td>
</tr>
<tr>
<td>DTMF</td>
<td>Dual Tone Multi Frequency</td>
</tr>
<tr>
<td>EC</td>
<td>Echo Cancellation</td>
</tr>
<tr>
<td>FSK</td>
<td>Frequency Shift Keying</td>
</tr>
<tr>
<td>FXO</td>
<td>Foreign Exchange Office</td>
</tr>
<tr>
<td>FXS</td>
<td>Foreign Exchange Subscriber</td>
</tr>
<tr>
<td>GPL</td>
<td>General Public License</td>
</tr>
<tr>
<td>HSS</td>
<td>High Speed Serial</td>
</tr>
<tr>
<td>IDS</td>
<td>Infrastructure DSP Solution</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>LSP</td>
<td>Linux Support Package</td>
</tr>
<tr>
<td>Term</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>NPE</td>
<td>Network Processor Engine</td>
</tr>
<tr>
<td>PLC</td>
<td>Packet Loss Concealment</td>
</tr>
<tr>
<td>RFC</td>
<td>Request For Comments</td>
</tr>
<tr>
<td>RTP</td>
<td>Real-time Transport Protocol</td>
</tr>
<tr>
<td>SLIC</td>
<td>Subscriber Line Interface Circuit</td>
</tr>
<tr>
<td>SRTP</td>
<td>Secure RTP</td>
</tr>
<tr>
<td>TDM</td>
<td>Time Division Multiplexing</td>
</tr>
<tr>
<td>USCI</td>
<td>Unified Speech Codec Interface</td>
</tr>
<tr>
<td>VAD</td>
<td>Voice Active Detection</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice Over IP</td>
</tr>
</tbody>
</table>

### 1.2 Reference Documents

<table>
<thead>
<tr>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Intel® IXP400 Software Version 2.3 Software Release Notes</em></td>
</tr>
<tr>
<td><em>Intel® IXP400 Software Version 2.1 Software Release Notes</em></td>
</tr>
<tr>
<td><em>Intel® Infrastructure DSP Solution Version 1.2 Codelet Demo Guide</em></td>
</tr>
<tr>
<td><em>Intel® Infrastructure DSP Solution Version 1.2 API Reference Manual</em></td>
</tr>
<tr>
<td><em>Intel® Infrastructure DSP Solution Version 1.2 Programmer’s Guide</em></td>
</tr>
</tbody>
</table>
2 Overview

The Intel® Infrastructure DSP Solution Version 1.2 Pre-Production Release 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 Intel® IXDP465 Development 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 2.3/2.1).

The following summarizes the software and hardware requirements for development:

- **Software**
  - Intel® IXP400 Software Version 2.3 in case of MVL 4.0 and kernel 2.6
  - Intel® IXP400 Software Version 2.1 in case of MVL 3.1 and kernel 2.4

- **Linux**
  - Refer to Intel® IXP400 Software Version 2.3 Software Release Notes for requirements on Linux Support Package
  - Refer to Intel® IXP400 Software Version 2.1 Software Release Notes for requirements on Linux Support Package

- **Platforms**
  - Intel® IXDP465 Development Platform
  - The Intel® Infrastructure DSP Solution Version 1.2 Pre-Production Release does not support Intel® IXP435 Multi-Service Residential 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:

- T.38 Fax as pluggable module with Illico® based data pump V.17, V.29, V.27Ter and V.21 fax modulation /demodulation
- Enhanced 128-ms Line echo cancellation for narrowband
- Line echo cancellation 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
• 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.
• 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. It reports T30 preamble events on both Tx and Rx direction separately.
• APIs in NET component for Tone Disabler 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 & Rx direction.
• 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 and Echo Return Loss Enhancement (ERLE).
• Audio mixer for three-way calls
• 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 Acoustic Echo Cancellation.
3 Release Components

3.1 Software Components

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

```
IDS
  Makefile
  drivers
  include
  plugin
  lib
  IXP
    Codec
      libg722.a
      libg723.1.a
      libg726.a
      libg729i.a
      libillico.a
      libippscs2.a
      libippss2.a
      libt38.a
    Foundation
      libdsp.a
      libosal.a
      liblec.a

    2.4
    Codec
      libg722.a
      libg723.1.a
      libg726.a
      libg729i.a
      libillico.a
      libippscs2.a
      libippss2.a
      libt38.a
    Foundation
      libdsp.a
      libosal.a
      liblec.a
```

3.1.1 Makefile

The following files are in the top-level directory:

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Makefile</td>
<td>Intel® IXP4XX Product Line Software Release top-level Makefile with updates to enable DSP-codelet related source code.</td>
</tr>
</tbody>
</table>
### 3.1.2 Header Files

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

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FWTypes.h</td>
<td>Contains type definitions.</td>
</tr>
<tr>
<td>msgdef.h</td>
<td>Contains message header definitions and function prototypes for sending control messages and receiving reply messages and event notifications.</td>
</tr>
<tr>
<td>dspcis.h</td>
<td>The core API. This file is the component interface specification.</td>
</tr>
<tr>
<td>dspcfg.h</td>
<td>Contains the API prototypes to configure and initialize the software module.</td>
</tr>
<tr>
<td>dspcfg2.h</td>
<td>Additional API prototypes to configure and initialize the software module.</td>
</tr>
<tr>
<td>dspitone.h</td>
<td>Contains the API prototypes to add new tone definitions.</td>
</tr>
<tr>
<td>ToneJP.h</td>
<td>Contains the definition of Japan call-progress-tone IDs.</td>
</tr>
<tr>
<td>ToneUS.h</td>
<td>Contains the definition of U.S. call-progress-tone IDs.</td>
</tr>
<tr>
<td>TonePRC.h</td>
<td>Contains the definition of China call-progress-tone IDs.</td>
</tr>
<tr>
<td>SCError.h</td>
<td>Contains the common error code.</td>
</tr>
<tr>
<td>HssAcc.h</td>
<td>Contains type definitions for HSS access layer</td>
</tr>
<tr>
<td>IxHssAcc.h</td>
<td>Contains type definitions for HSS access layer</td>
</tr>
<tr>
<td>IxIDS_SlicCodec.h</td>
<td>Contains the public APIs of the FXS/FXO devices.</td>
</tr>
<tr>
<td>usc_base.h</td>
<td>Defines USC base function APIs</td>
</tr>
<tr>
<td>usc_ec.h</td>
<td>Defines USC echo canceller APIs</td>
</tr>
<tr>
<td>usc.h</td>
<td>Defines USC speech codec APIs</td>
</tr>
<tr>
<td>usc_t38.h</td>
<td>Defines USC T.38 fax APIs</td>
</tr>
</tbody>
</table>

### 3.1.3 Plug-in Configuration Files

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

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PlugInConfig.h</td>
<td>This is mainly used to define all plug-in header files.</td>
</tr>
<tr>
<td>PlugInConfig.c</td>
<td>Configure the speech CODECs/EC plug-ins. Adds entry for either codec or EC (any order) plug-ins in the plug-in list.</td>
</tr>
<tr>
<td>PlugInDefs.h</td>
<td>Contains various plug-in interface types.</td>
</tr>
</tbody>
</table>
3.1.4 Binary Files for Linux*

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>libdsp.a, libosal.a</td>
<td>This is the core library, present in directory for kernel 2.6</td>
</tr>
<tr>
<td></td>
<td>&lt;workdir&gt;/IDS/lib/IXP/2_6/Foundation/ for kernel 2.4</td>
</tr>
<tr>
<td></td>
<td>&lt;workdir&gt;/IDS/lib/IXP/2_4/Foundation/</td>
</tr>
<tr>
<td>liblec.a</td>
<td>This is Line Echo Canceller library, present in directory</td>
</tr>
<tr>
<td></td>
<td>&lt;workdir&gt;/IDS/lib/IXP/2_6/Foundation/ for kernel 2.4</td>
</tr>
<tr>
<td></td>
<td>&lt;workdir&gt;/IDS/lib/IXP/2_4/Foundation/</td>
</tr>
<tr>
<td></td>
<td>(Note: Acoustic Echo Canceller is not supported in this version)</td>
</tr>
<tr>
<td>libippscs2.a, libipps2.a,</td>
<td>These are plug-in speech CODECs provided by Intel, present in directory</td>
</tr>
<tr>
<td>libg722.a, libg723.1.a,</td>
<td>for kernel 2.6</td>
</tr>
<tr>
<td>libg726.a, libg729i.a</td>
<td>&lt;workdir&gt;/IDS/lib/IXP/2_6/Codec</td>
</tr>
<tr>
<td></td>
<td>for kernel 2.4</td>
</tr>
<tr>
<td></td>
<td>&lt;workdir&gt;/IDS/lib/IXP/2_4/Codec</td>
</tr>
<tr>
<td>libt38.a, libillico.a</td>
<td>These are T38 fax plug-in component provided by Intel, present in directory</td>
</tr>
<tr>
<td></td>
<td>for kernel 2.6</td>
</tr>
<tr>
<td></td>
<td>&lt;workdir&gt;/IDS/lib/IXP/2_6/Codec</td>
</tr>
<tr>
<td></td>
<td>for kernel 2.4</td>
</tr>
<tr>
<td></td>
<td>&lt;workdir&gt;/IDS/lib/IXP/2_4/Codec</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLIC</td>
<td>Source files for the FXS/FXO device driver</td>
</tr>
<tr>
<td>HSS</td>
<td>Source files for the HSS driver</td>
</tr>
<tr>
<td>dspApp</td>
<td>Source files for Linux*-specific DSP user-mode application</td>
</tr>
</tbody>
</table>
4 Installing and Building the Software

Intel Infrastructure DSP Solution Version 1.2 runs over Linux* operating systems with

- Intel® IXP400 Software Version 2.3 for kernel 2.6
- Intel® IXP400 Software Version 2.1 for kernel 2.4

Prerequisite software packages:

- Refer to Intel® IXP400 Software Version 2.3 Software Release Notes Table 2. or
- Refer to Intel® IXP400 Software Version 2.1 Software Release Notes Table 2.

4.1 Linux*

4.1.1 Requirements for Infrastructure DSP Solution

The table below lists the files needed to build Infrastructure DSP Solution. These requirements assume that prerequisite software packages stated in Intel® IXP400 Software Version 2.3 or version 2.1 in case of kernel 2.4 Software Release Notes Table 2 is available for installation.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSD_IDS_HSS_Drv_V1_2.zip</td>
<td>Intel® Infrastructure DSP Solution Version 1.2 release with HSS driver,</td>
</tr>
<tr>
<td></td>
<td>SLIC driver, Codec Library, Codelets, Foundation Library and patches.</td>
</tr>
<tr>
<td>BSD_IDS_SLIC_Drv_V1_2.zip</td>
<td></td>
</tr>
<tr>
<td>IPL_IDS_CodecsLibrary_V1_2.zip</td>
<td></td>
</tr>
<tr>
<td>IPL_IDS_Codelets_V1_2.zip</td>
<td></td>
</tr>
<tr>
<td>IPL_IDS_FoundationLibrary_V1_2.zip</td>
<td></td>
</tr>
</tbody>
</table>

4.1.2 Installing the Software

4.1.2.1 Prerequisites

Refer to Intel® IXP400 Software Version 2.3/2.1 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 2.3

Refer to Intel® IXP400 Software Version 2.3 Software Release Notes Section 6.4, from step 1 to step 11 for the installation procedures. Step 1 to step 12 of Section 6.4 installation procedures includes:
**Intel® Infrastructure DSP Solution**

- Created working directory, `<workdir>` (step 1)
- Setup LSP kernel source tree (step 2 and step 3)
- Apply the kernel patch (step 4)
- Extracted Intel® IXP400 Software Version 2.3 zip files (step 5)
- Extracted NPE Microcode Image zip file (step 6)
- Configured and setup environment variables (step 7 and step 8). Refer to Intel® IXP400 Software Version 2.3 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 10, step 11, and step 12)

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

<table>
<thead>
<tr>
<th>Intel® IXDP465 Development Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>(33.330 MHz OSC)</strong></td>
</tr>
</tbody>
</table>

```
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.
3. Unzip the zip files by giving the following commands:

```bash
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:

```bash
<work_dir>
    IDS
    xip400_xscale_sw
    xip_osal
    linux
```

### 4.1.3 Installing SRTP Source Code

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

```bash
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 0). Make the following changes in the Makefile as below:

```bash
cd <workdir>/srtp
./configure
vi Makefile
```

Change the `cc make file macro assignment to`
as cc=gcc  

to  

cc= /opt/montavista/pro/devkit/arm/xscale_be/bin/xscale_be-gcc  

In case of MontaVista Linux (MVL) 4.0, in addition to above changes, add the following additional flag  

CFLAGS += -mabi=apcs-gnu  

### 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 2.3/2.1 Software Release Notes Section 5.2 step 6) should be maintained.

**Environment variables for Kernel 2.6**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINUX_SRC</td>
<td><code>&lt;workdir&gt;/linux</code></td>
</tr>
<tr>
<td>export LINUX_SRC</td>
<td></td>
</tr>
<tr>
<td>IX_CONTROLLED_COUNTRY_BUILD</td>
<td>1</td>
</tr>
<tr>
<td>export IX_CONTROLLED_COUNTRY_BUILD</td>
<td></td>
</tr>
<tr>
<td>MYCSR_BASE</td>
<td><code>$IX_XSCALE_SW</code></td>
</tr>
<tr>
<td>export MYCSR_BASE</td>
<td></td>
</tr>
<tr>
<td>TOOLENV</td>
<td><code>linux</code></td>
</tr>
<tr>
<td>export TOOLENV</td>
<td></td>
</tr>
<tr>
<td>MYDSR_BASE</td>
<td><code>&lt;workdir&gt;/IDS</code></td>
</tr>
<tr>
<td>export MYDSR_BASE</td>
<td></td>
</tr>
</tbody>
</table>
SRTP=1

**Note**: Only include this environment variable if SRTP is used, else do not define this environment variable.

```plaintext
SRTP_BASE = <workdir>/srt
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_ENV= mvl4.0
export IX_ENV

IX_XSCALE_SW_VER=23
export IX_XSCALE_SW_VER
```

Intel® IXP400 Software Version 2.1 Software Release Notes Section 5.2 step 6) should be maintained.

**Environment variables for Kernel 2.4**

```plaintext
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

IX_LINUXVER=2.4
Export IX_LINUXVER

IX_ENV= mvl3.1
export IX_ENV

IX_XSCALE_SW_VER=21
export IX_XSCALE_SW_VER
```
## 4.1.4.2 Edit PlugInConfig.c

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

<table>
<thead>
<tr>
<th>PlugInConfig.c</th>
</tr>
</thead>
<tbody>
<tr>
<td>#include &quot;PlugInConfig.h&quot; /* The global PlugInList*/</td>
</tr>
<tr>
<td>PLUG_IN_LIST_BEGIN</td>
</tr>
<tr>
<td>/<em><strong><strong><strong><strong>Don't edit or modify above this line</strong></strong></strong></strong></em>/</td>
</tr>
<tr>
<td>/*Add plug-in entries below */</td>
</tr>
<tr>
<td>USCI_CODEC_PLUG_IN (USC_G722SB_Fxns,XCODER_TYPE_G722,0)</td>
</tr>
<tr>
<td>USCI_CODEC_PLUG_IN (USC_G729A_Fxns,XCODER_TYPE_G729A,0)</td>
</tr>
<tr>
<td>USCI_CODEC_PLUG_IN (USC_G726_Fxns,XCODER_TYPE_G726_40,0)</td>
</tr>
<tr>
<td>USCI_CODEC_PLUG_IN (USC_G726_Fxns,XCODER_TYPE_G726_32,0)</td>
</tr>
<tr>
<td>USCI_CODEC_PLUG_IN (USC_G726_Fxns,XCODER_TYPE_G726_24,0)</td>
</tr>
<tr>
<td>USCI_CODEC_PLUG_IN (USC_G726_Fxns,XCODER_TYPE_G726_16,0)</td>
</tr>
<tr>
<td>USCI_CODEC_PLUG_IN (USC_G723_Fxns,XCODER_TYPE_G723,0)</td>
</tr>
<tr>
<td>/<em>Echo canceller plug_ins</em>/</td>
</tr>
<tr>
<td>USCI_EC_PLUG_IN (USC_LECINT_Fxns,XEC_TYPE_LINE_EC,0)</td>
</tr>
<tr>
<td>/<em>T38 Fax plug-in</em>/</td>
</tr>
<tr>
<td>USCI_FAX_PLUG_IN (USC_T38INT_Fxns,XFAX_TYPE_T38,0)</td>
</tr>
<tr>
<td>/<em><strong><strong><strong><strong>Don't edit or modify beyond this line</strong></strong></strong></strong></em>/</td>
</tr>
<tr>
<td>PLUG_IN_LIST_END</td>
</tr>
</tbody>
</table>

**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:

```
#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 "lecinterface.h"  /*Line echo canceller*/
```
Note: Acoustic Echo Canceller (#include “ecusc_int.h”) and G.729.1 codec (#include “g729_lusc.h”) are not supported in this release. It is recommended that both plug-ins should be commented.

Examples

Case 1

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

```c
#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

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

and comment all other lines between the lines DECLARE_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-40Kbps and line echo canceller, then select the following two lines:

```c
#include "g726usc.h"
#include "lecinterface.h"
```

and comment all other lines between the lines DECLARE_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:

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

2. To build bootable kernel image, *zImage*, refer to Intel® IXP400 Software Version 2.3 Software Release Notes Section 6.5 step 1 to step 4. This will build and copy *zImage* into /tftpboot/ directory.

3. To build Ethernet Kernel Module, *ixp400_eth.ko*, refer to Intel® IXP400 Software Version 2.3 Software Release Notes Section 6.6.1 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 2.3, refer to Intel® IXP400 Software Version 2.3 Software Release Notes Section 6.6.2. 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:

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

4.1.4.6 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.

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

This creates *libplug.a* and is available in the directory <workdir>/IDS/lib.
4.1.4.7 Building DSP Application

Build DSP application by giving the \texttt{make app} command. This builds the application “IxDspCodeletApp”. This executable contains the plug-in components selected in PlugInConfig.c file.

\begin{verbatim}
  cd <workdir>/IDS
  make app
\end{verbatim}

The executable IxDspCodeletApp is created in the directory \texttt{<workdir>/IDS/lib}.

4.1.4.8 Building SLIC Driver

\begin{verbatim}
  cd <workdir>/IDS
  make slic
\end{verbatim}

This creates \texttt{ixp400_codec.ko} and is available in the directory \texttt{<workdir>/IDS/lib}.

4.1.4.9 Building HSS Driver

\begin{verbatim}
  cd <workdir>/IDS
  make hssdriver
\end{verbatim}

This creates \texttt{hssdriver.ko} and is available in the directory \texttt{<workdir>/IDS/lib}.
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:

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/montavista4.0/montavista/pro/devkit/arm/xscale_be/target ";
    }
}
```

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
}
```

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/montavista4.0/montavista/pro/devkit/arm/xscale_be/target *(rw,no_root_squash)
```
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

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 Intel® IXDP465 Development Platform:
1. Fix four-ports-voice-module to HSS0 port of Intel® IXDP465 Development Platform
2. Connect four telephones to FXS0 to FXS3 ports of voice module on Intel® IXDP465 Development Platform
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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run script at boot:</td>
<td>false</td>
</tr>
<tr>
<td>Use BOOTP for network configuration:</td>
<td>false</td>
</tr>
<tr>
<td>Local IP address:</td>
<td>192.168.0.50</td>
</tr>
<tr>
<td>Default server IP address:</td>
<td>192.168.0.100</td>
</tr>
<tr>
<td>Console baud rate:</td>
<td>115200</td>
</tr>
<tr>
<td>GDB connection port:</td>
<td>9000</td>
</tr>
<tr>
<td>Network debug at boot time:</td>
<td>false</td>
</tr>
<tr>
<td>Update RedBoot non-volatile configuration – are you sure (y/n)?</td>
<td>y</td>
</tr>
</tbody>
</table>

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

### For MontaVista® Linux* 3.1 (Kernal 2.4)

- cd /tmp
- insmod ixp400.o
- mknod /dev/ixNpe c 241 0
- cat IxNpeMicrocode.dat > /dev/ixNpe
- insmod ixp400_eth.o
- insmod ixp400_codec.o
- insmod hssdriver.o
- mknod /dev/ixSlicModule c 252 0

### For MontaVista® Linux* 4.0 (kernal 2.6)

- 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

### MVL 4.0 files

<table>
<thead>
<tr>
<th>Filename</th>
<th>Available in directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>ixp400.ko</td>
<td>For Intel® IXDP465 Development Platform</td>
</tr>
<tr>
<td></td>
<td>&lt;workdir&gt;/ixp400_xscale_sw/ixp46X/lib/linuxbe</td>
</tr>
<tr>
<td>ixp400_eth.ko</td>
<td>&lt;workdir&gt;/linux/drivers/net</td>
</tr>
<tr>
<td>hssdriver.ko</td>
<td>&lt;workdir&gt;/IDS/drivers/HSS/lib</td>
</tr>
<tr>
<td>ixp400_codec.ko</td>
<td>&lt;workdir&gt;/IDS/drivers/SLIC/lib</td>
</tr>
<tr>
<td>IxDspCodeletApp</td>
<td>&lt;workdir&gt;/IDS/codelets/dspApp</td>
</tr>
<tr>
<td>IxNpeMicrocode.dat</td>
<td>&lt;workdir&gt;/ixp400_xscale_sw/ixp46X/lib/linuxbe</td>
</tr>
</tbody>
</table>

### For MVL 3.1 files

<table>
<thead>
<tr>
<th>Filename</th>
<th>Available in directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>ixp400.o</td>
<td>For Intel® IXDP465 Development Platform</td>
</tr>
<tr>
<td></td>
<td>&lt;workdir&gt;/ixp400_xscale_sw/ixp46X/lib/linuxbe</td>
</tr>
<tr>
<td>ixp400_eth.o</td>
<td>&lt;workdir&gt;/linux/drivers/net</td>
</tr>
<tr>
<td>hssdriver.o</td>
<td>&lt;workdir&gt;/IDS/drivers/HSS/lib</td>
</tr>
<tr>
<td>ixp400_codec.o</td>
<td>&lt;workdir&gt;/IDS/drivers/SLIC/lib</td>
</tr>
<tr>
<td>IxDspCodeletApp</td>
<td>&lt;workdir&gt;/IDS/codelets/dspApp</td>
</tr>
<tr>
<td>IxNpeMicrocode.dat</td>
<td>&lt;workdir&gt;/ixp400_xscale_sw/ixp46X/lib/linuxbe</td>
</tr>
</tbody>
</table>
mknod /dev/hssdriver c 251 0
ifconfig ixp0 192.168.10.1
ifconfig ixp1 192.168.20.1
chmod 777 IxDspCodeletApp
./IxDspCodeletApp

mknod /dev/hssdriver c 251 0
ifconfig ixp0 192.168.10.1
ifconfig ixp1 192.168.20.1
chmod 777 IxDspCodeletApp
./IxDspCodeletApp

Note:
If kernel module version mismatch is seen while doing insmod, specify “-f” option.
Ex: insmod -f ixp400_codec.o

Note:
If kernel module version mismatch is seen while doing insmod, specify “-f” option.
Ex: insmod -f ixp400_codec.ko

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set country code (1:US, 81:Japan, 86:China)</td>
<td>1</td>
</tr>
<tr>
<td>Enter companding mode (0:A-law, 1:u-law, 2:Linear)</td>
<td>1</td>
</tr>
<tr>
<td>Enter slic type (0:Narrow band,1:Wide band)</td>
<td>0</td>
</tr>
</tbody>
</table>
6 Demo ‘Screen Shot’

./IxDspCodeletApp

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
DLcid 11020001-01

********************************************************************
*       Intel Infrastructure DSP Solution       *
*       Release 1.2 (Pre-Production Release)*
*       Aug 18 2007, 23:27:03               *
*       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


Initializing encoders


Initializing Player decoders


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 -- SLIC is already initialised

-----
Number of TDM terminations = 4

Number of IP terminations = 8

Number of Player FXS Callback is already initialised.

- 5

Number of Mixers = 4

Number of ports per Mixer = 4

Country code = 1

- ---------------------------

- 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
7 Known Bugs/Open Issues

This is pre-production release, not all features are fully validated.

7.1 T38 Fax plug-in module issues

Reference Number: IXA00178271

Intel® IXP400 Software Version(s): 2.3/2.1

Platform(s): IXDP465

Description: In T.38 Fax plug-in module tested with

- V.27ter 2400 bps Nominal profile “TX 3 Pg Best ECM Best Enc V.27ter 2.4k Best Res” fails intermittently.
- V.17 14400 bps Nominal profile “TX 3 Pg Best ECM Best Enc Best Mod 400x400” fails intermittently.

All these tests are done using ChannelTrap (Quality logics® FaxLab) version 5.0

Implication: The Fax session may fail with modulation V.27ter 2.4k and V.17 14400

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

7.2 Improper Codelet SLIC menu ID’s

Reference Number: IXA00178274

Intel® IXP400 Software Version(s): 2.3/2.1

Platform(s): IXDP465

Description: In IDS codelet SLIC menu, menu ID’s from 17 to 22 are improper, As a workaround the correct usage of menu items are described in IDS 1.2 Codelet Demo Guide.

Implication: Improper menu operations.

Resolution/Workaround: Please contact your Intel representative for resolution or additional information.
7.3 **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.

7.4 **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.
7.5 Improper RFC2833 tones communication between channels when SRTP is enabled

Reference Number: IXA00165500

Intel® IXP400 Software Version(s): 2.4

Platform(s): Intel® IXP400, Intel® IXP425 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.

7.6 Unexpected events reporting of Phase Reversal and Silence detection

Reference Number: IXA00165503

Intel® IXP400 Software Version(s): 2.4

Platform(s): Intel® IXP400, Intel® IXP425 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: The work around solution is to set EC tone Disabler Silence Period (XPARMID_NET_ECTDSILENCE TIME) from codelet prompt or API. The display parameter shows it as 250ms which is default value. The user has to set this value to required value or default value (250ms) using the codelet menu or using API call in their application. This is required as this parameter is not initialized to default value during initialization.