

# Intel® Infrastructure DSP Solution Version 1.2

Software Release Notes

---

*December 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 Intel® IXP4XX Product Line of 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 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 .....	9
	3.1 Software Components .....	9
	3.1.1 Makefile .....	9
	3.1.2 Header Files .....	10
	3.1.3 Plug-in Configuration Files .....	10
	3.1.4 Binary Files for Linux* .....	11
	3.1.5 Drivers and DSP Codelet .....	11
4	Installing and Building the Software.....	12
	4.1 Linux* .....	12
	4.1.1 Requirements for the Intel® Infrastructure DSP Solution .....	12
	4.1.2 Installing the Software.....	12
	4.1.3 Installing SRTP Source Code .....	14
	4.1.4 Building the Software.....	14
5	Testing the Software in Linux* .....	21
6	Demo Screen Captures for the Intel® IXP435 Multi-Service Residential Gateway Reference Platform.....	24
7	Limitations .....	28
	7.1 T38 Fax Plug-in Module Issues .....	28
	7.2 Fluctuation of CPU Utilization with “top” Command Measurement Reference..	28
	7.3 No Voice Path If Hssdriver and Codec Driver are Reinserted into Kernel.....	29
	7.4 Low Voice Quality in Packet Loss Testing .....	29
	7.5 T.38 Interoperability Test with Third Party T.38 Gateway .....	30



## Revision History

---

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



# 1 Introduction

---

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® IXP43X 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

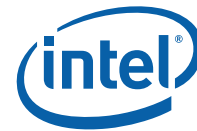
Term	Description
AGC	Automatic Gain Control
ALC	Automatic Level Control
API	Application Programming Interface
CNG	Comfort Noise Generation
CODEC	Coder Decoder
DP	Development Platform
DSCP	Differentiated service Code Point
DSP	Digital Signal Processing
DTMF	Dual Tone Multi Frequency
EC	Echo Cancellation
FSK	Frequency Shift Keying
FXO	Foreign Exchange Office
FXS	Foreign Exchange Subscriber
GPL	General Public License
HSS	High Speed Serial
iLBC	internet Low Bit rate Codec



Term	Description
IP	Internet Protocol
LSP	Linux Support Package
NPE	Network Processor Engine
PLC	Packet Loss Concealment
QoS	Quality of Service
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 1.2 API Reference Manual</i>
<i>Intel® Infrastructure DSP Solution Version 1.2 Programmer's Guide</i>
<i>Intel® Infrastructure DSP Solution Version 1.2 Codelet Demo Guide</i>
<i>Intel® IXP400 Software Version 3.0 Software Release Notes</i>



## 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 have been targeted to work on Intel® IXP435 Multi-Service Residential 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 3.0).

The following summarizes the software and hardware requirements for development:

- Software
  - Intel® IXP400 Software Version 3.0
- Linux
  - Refer to Intel® IXP400 Software Version 3.0 Software Release Notes for requirements on Linux Support Package
- Platforms
  - 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:

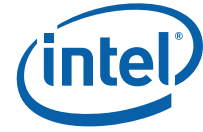
- Line echo cancellation up to 128 ms for narrowband or up to 64 ms for wideband
- G.711  $\mu$ -law and A-law CODEC with VAD and CNG support
- G.723.1 and G.729ab with VAD and CNG support
- G.726 with 16, 24, 32 and 40 Kbps rates and RFC3551 and I.366 Annex E packing formats
- G.722 wideband codec
- G.729.1 Wide band as well as Narrow band 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<sup>1</sup>
  - 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
  - Support for DSCP QoS
- This release does not support Acoustic Echo Cancellation.

---

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

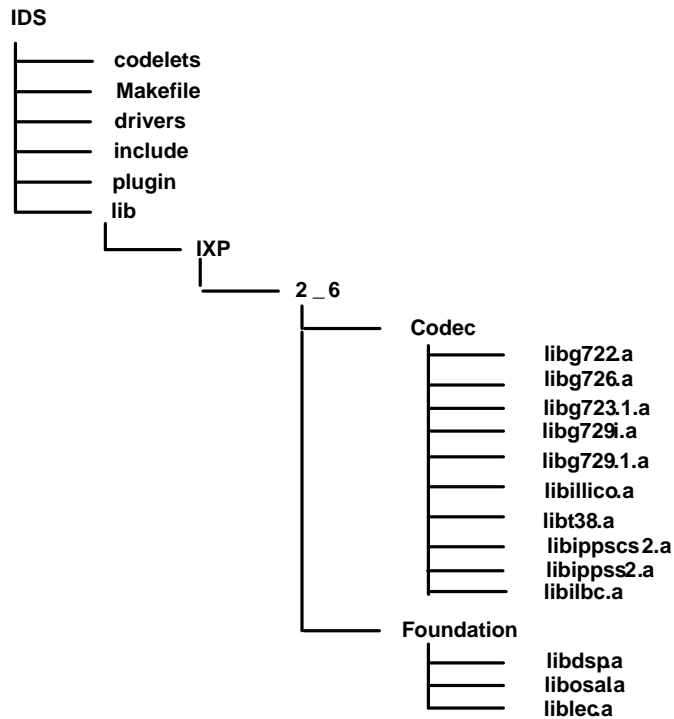


### 3 Release Components

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:



##### 3.1.1 Makefile

The following files are in the top-level directory:

Component	Description
Makefile	Intel® IXP43X 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
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 plugin header files.
PlugInConfig.c	Configure the speech CODECs/EC plugins. Adds entry for either codec or EC (any order) plug-ins in the plugin list.



Component	Description
PlugInDefs.h	Contains various plugin 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/
libosal.a	Operating Systems Abstraction Layer Library for User Space, 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/ <b>(Note:</b> 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 <workdir>/IDS/lib/IXP/2_6/Codec
libt38.a, libillico.a	These are T38 fax plug-in component 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 1.2 runs over Linux\* operating system with Intel® IXP400 Software Version 3.0.

Prerequisite software packages:

- Refer to *Intel® IXP400 Software Version 3.0 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 3.0 Software Release Notes* Table 2 are 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

#### 4.1.2 Installing the Software

##### 4.1.2.1 Prerequisites

Refer to *Intel® IXP400 Software Version 3.0 Software Release Notes* Section 5.1 to install the desired development/reference platform LSP and setup host development system.

##### 4.1.2.2 Installing Intel® IXP400 Software Version 3.0

Refer to *Intel® IXP400 Software Version 3.0 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 3.0 zip files (step 4)



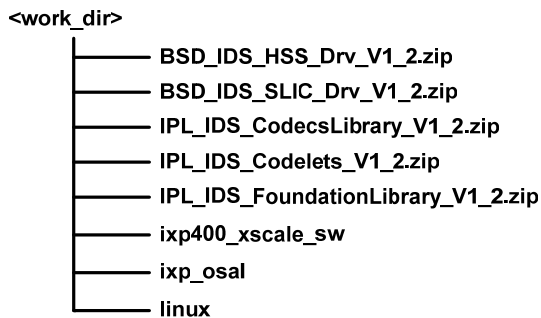
- Extract NPE Microcode Image zip file (step 5)
- Configure and setup environment variables (step 6, step 7, and step 8). Refer to *Intel® IXP400 Software version 3.0 Software Release Notes* Table 5 to set the appropriate values for environment variables.
- Patch Intel® IXP400 Software Linux Ethernet Driver (step 9, step 10, and step 11)

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® IXP435 Reference Platform, the `CLOCK_TICK_RATE` settings are as shown in the following table:

<b>Intel® IXP435 Reference Platform (33.330 MHz OSC)</b>
<pre>linux/include/asm-arm/arch-ixp4xx/timex.h #define  FREQ  66660000</pre>

#### 4.1.2.3 Installing 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.



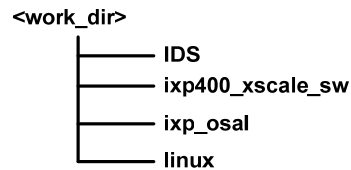
3. Unzip the zip files by giving the following commands:

```
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:



### 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.

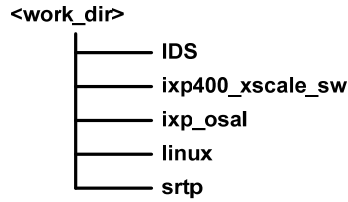
```
cd <workdir>
gunzip -c srtp-1.4.1.tar | tar -xf -
```

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 3.0 Software Release Notes* Section 5.2 Step 6) should be maintained.

<pre>LINUX_SRC=&lt;workdir&gt;/linux export LINUX_SRC</pre>
<pre>IX_CONTROLLED_COUNTRY_BUILD=1 export IX_CONTROLLED_COUNTRY_BUILD</pre> <p><b>Note:</b> Include this environment variable only if you do not use the crypto version of the Access Library Software</p>
<pre>MYCSR_BASE=\$IX_XSCALE_SW export MYCSR_BASE</pre>
<pre>TOOLENV=linux export TOOLENV</pre>
<pre>MYDSR_BASE=&lt;workdir&gt;/IDS export MYDSR_BASE</pre>
<pre>SRTP=1</pre> <p><b>Note:</b> Only include this environment variable if SRTP is used, else do not define this environment variable.</p>
<pre>SRTP_BASE=&lt;workdir&gt;/srtp export SRTP_BASE</pre> <p><b>Note:</b> 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>
<pre>IX_XSCALE_SW_VER=30 export IX_XSCALE_SW_VER</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.

#### 4.1.4.2 Edit PlugInConfig.c

Extract of the 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 */
/*Codec plug-ins*/

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)
/*T38 Fax plug-in*/
USCI_FAX_PLUG_IN(USC_T38INT_Fxns,XFAX_TYPE_T38,0)
USCI_CODEEC_PLUG_IN(USC_G7291_Fxns,XCODER_TYPE_G729_1,0)
USCI_CODEEC_PLUG_IN(USC_ILBC_Fxns,XCODER_TYPE_ILBC_30MS,0)
USCI_CODEEC_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_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 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 "G726, rate-40Kbps" and "line echo canceller", then select the following two lines

```
USCI_CODEEC_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" </pre>



```
#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_lusc.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 this plug-in 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 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.

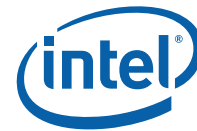
This example shows how to build Line Echo Canceller plug-in only. This example demonstrates that the application requires default G.711 u-law and G.711 a-law codecs with Line Echo Canceller. All other codecs will not be included in this case.

#### Case 3

If you want to select G,726, rate-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 3.0 Software Release Notes* Section 5.3 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 3.0 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 3.0*, refer to *Intel® IXP400 Software Version 3.0 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 Section 4.1.4.7 to Section 4.1.4.9 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 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 `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 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 ixp435 {
        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 is based on the assumption that the host system and Intel® IXP435 Multi-Service Residential Gateway Reference Platform is connected via a cross-over Ethernet cable using the Intel® 8255x-based PCI card in the target platform.

For Intel® IXP435 Reference Platform:

1. The FXS/FXO ports are available on the platform by default and external voice module is not required.
2. There are only two FXS ports; so only 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
<pre>insmod ixp400.ko mknod /dev/ixNpe c 241 0 cat IxNpeMicrocode.dat &gt; /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>

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

Parameter	Value
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

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



## 6 Demo Screen Captures for the 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
ixSlic_init_module :: LOADED SUCCESSFULLY IXSLIC MODULE ...
hssdriver_init_module :: LOADED SUCCESSFULLY HSS Driver ...
Set country code (1:US, 81:Japan, 86:China) - 1

Existing NPEA Image ID: 0x109E0000

New NPEA Image ID: 0x109E0000
---- Initializing HSS Device ----
Enter companding mode (0:A-law, 1:u-law, 2:Linear) - 1
Enter slic type (0:Narrow band,1:Wide band) - 0
DLcid 01020000-09
*****
* Intel Infrastructure DSP Solution *
* Release 1.2 *
* Dec 17 2007, 20:01:15 *
* 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_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 1: T38 Fax Modem    - Done
Channel 2: T38 Fax Modem    - Done
Channel 3: T38 Fax Modem    - 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

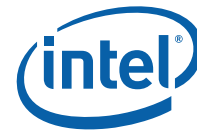
*** DSP Codelet Socket And Voice Channel Config Info ***

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 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  
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  
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)>>
```



## 7 Limitations

---

### 7.1 T38 Fax Plug-in Module Issues

**Reference Number:** IXA00178271

**Intel® IXP400 Software Version(s):** 3.0

**Platform:** Intel® IXP435 Multi-Service Residential Gateway Reference Platform

**Description:** 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 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

**Number:** IXA00153759

**Intel® IXP400 Software Version(s):** 3.0

**Platform:** Intel® IXP435 Multi-Service Residential 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 No Voice Path If Hssdriver and Codec Driver are Reinserted into Kernel

**Reference Number:** IXA00178563

**Intel® IXP400 Software Version(s):** 3.0

**Platform(s):** Intel® IXP435 Multi-Service Residential Gateway Reference Platform

**Description:** When you load the kernel loadable modules hssdriver.ko and ixp400\_codec.ko and start the application with normal options, voice path exists. When you Exit the application and use "rmmod" to remove the hssdriver and ixp400\_codec drivers from the kernel, and once again re-insert using "insmod", and restart the application, there is no voice path.

**Implication:** There is no issue while using for the first time.

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

## 7.4 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% random IP Packet loss test environment using IP Cloud\*, whenever there are two consecutive packet losses, voice quality is low and voice quality recovers in a few seconds. This limitation is observed with G.729ab, G.729.1, G.723.1, and iLBC codecs only.

**Implication:** G.729ab, G.729.1, G.723.1, and iLBC codecs have low voice quality with random 5% or more IP Packet loss. Voice quality will recover in a few seconds.

**Resolution/Workaround:** No workaround or resolution. This is a limitation for G.729ab, G.729.1, G.723.1, and iLBC codecs under 5% packet loss testing. This is not a defect.



## 7.5 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:** During T.38 Interoperability Test, the following results are observed:

- 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 third party T.38 gateway.

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