



Intel® IXA SDK Tools 3.51 Software Product Release Notes

(Product names and code names are subject to change without notice)

Use of Software Development Tools on the Intel® IXA SDK Tools 3.51 CD-ROM is subject to the terms of the INTEL SOFTWARE LICENSE AGREEMENT on the IXA SDK 3.51 CD-ROM. Before using such Tools, you must agree to accept the terms of the INTEL SOFTWARE LICENSE AGREEMENT.

Copyright © 2004, Intel Corporation. All Rights Reserved.

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

This product includes code licensed from RSA Security, Inc. and some portions licensed from IBM are available at <http://oss.software.ibm.com/icu4j/>.

This version of the Release Notes is up-to-date as of the time of the release. For the latest list of known software problems, please refer to the IXA SDK 3.51 Software Problem Listing that can be found online via Electronic Design Kit (EDK) on Intel Business Link (IBL). If you have an IBL account set up, please access the Network Processing EDK through https://teal.intel.com/scripts-edk/viewer/UI_CLCatalog.asp?edkId=2184. You can also subscribe to this EDK for easier access by clicking on "Add To My EDKs".

These product release notes are divided into the following sections:

- Installation and build procedures
- Patch installation procedures
- New features
- Major bugs fixed
- Known problems and workaround
- Compatibility notes
- Documentation issues
- File changes from the Intel® IXA SDK 3.5 to the Intel® IXA SDK 3.51

Installation and Build Procedures

1. The Intel® IXA SDK 3.1 and the Intel® IXA SDK 3.51 can co-exist on the same development system. However, it is recommended that you uninstall any previous version of the Intel® IXA SDK 3.1 before you install the Intel® IXA SDK 3.51 kit. Removing files does not clean up the registry settings so the uninstall is required. To uninstall: go to Start -> Settings -> control panel -> Add/Remove Software, click on IXA SDK 3.1 and click on Change/Remove to remove the current IXA SDK install.
2. To install the Intel® IXA SDK Tools 3.51 kit on a Windows 2000* system, run the Setupwin32.exe program located in the top-level directory of the CD-ROM, and follow the instructions. You need to accept the INTEL LICENSE AGREEMENT. It is recommended that you accept the default top-level directory location (C:\IXA_SDK_3.5). If you already have the Intel® IXP12DE product installed on your system, do not install the Intel® IXA SDK 3.51 on top of the Intel® IXP12DE directory structure.

NOTE: You must have administrative rights on the Windows system to perform the Intel® IXA SDK install.

3. The Intel® IXA SDK Tools 3.51 kit needs to be installed before installing other optional the Intel® IXA SDK CD kits.
4. Both the Intel® IXA SDK Tools 3.51 kit and the Intel® IXP2850 Simulation Environment CD need to be installed if your hardware target is the Intel® IXDP2850 Advanced Development Platform.



5. To install the Intel[®] IXA SDK Tools 3.51 kit on Linux Redhat* 7.3 system, change the directory to /opt and untar the `ixa_sdk_3.5.tgz` from the top-level directory of the CD-ROM.
6. There is a possibility that during installation text messages may not be displayed on the screen. You might just see buttons (like “next” and “accept”) to click and proceed with installation. If this happens then proceed with clicking the next button and the installation will take its normal course and complete successfully. This is a highly improbable case and has not been reproduced except in an isolated case.

New Features for the SDK 3.51

IXP2400 and IXP2800 Transactor

Added datawatch support, through the Workbench, for the Intel XScale[®] core, Thread CSRs, and PCI CSRs when in simulation mode.

Developer Workbench

1. Minor modifications have been made to Workbench toolbar buttons bitmaps.
2. Support for IXP2800 A0 and IXP2850 A0 has been dropped. If the user opens a project that was created for either of these chip revisions, the Workbench will automatically upgrade them for revision A1. A message box is displayed to notify the user. Similarly, support for IXP2400 A1 is dropped and projects are automatically upgraded to revision B0.
3. Simulation Performance Statistics now has a button for dumping all performance statistics to a comma separated value file. The file can be viewed with an editor or a tool that is capable of parsing the csv formatted file, such as Microsoft Excel.
4. Simulation Performance Statistic format has been changed to add additional digits of precision. Statistics are reports in percentages and in cycles.
5. Packet Simulation Status now has a button for dumping all performance statistics to a comma separated value file. The file can be viewed with an editor or a tool that is capable of parsing the csv formatted file, such as Microsoft Excel.
6. The maximum number of cycles of history that can be collected has been increased from 10,000 to 1,000,000. When the history cycle limit has been reached, oldest cycles are trimmed. Note that simulation and GUI performance can degrade significantly if you try to collect too many cycles on a computer with a slow CPU (< 1GHz) and little memory (512Mb or less). With an adequately sized page file, such a system can easily collect between 200,000 to 500,000 cycles, but excessive paging occurs above about 500,000 cycles collected. Experiments with typical microcode reference designs have produced a range of between 100 and 1000 bytes stored history per cycle. A 3GHz/1Gb system can easily handle the full 1,000,000 cycles collected for a typical microcode application.
7. Support for conditional breakpoints and breakpoint functions has been added. Breakpoint functions can be declared in script files, on the transactor command line, through the Workbench breakpoint properties dialog, and in user-written foreign models. A breakpoint function is called by the transactor when its associated instruction is about to be executed. If the function returns non-zero, the breakpoint occurs. If the function returns zero, program execution continues as if no breakpoint were present. Conditional breakpoints and breakpoint functions are particularly useful for instrumenting microcode applications, logging application performance, and pausing program execution if error conditions are detected.
8. In microengine c projects, switching between source view and list view while the cursor is in an inline function now maintains the correct line of code relationship between the two views.

Hardware Performance Monitor

1. In the Workbench, under hardware debugging, there is a new option to access the Performance Monitoring Unit (PMU) in the hardware. See “Known Problems and Workarounds” below for details on current limitations.
2. Supports the IXP2400 and IXP2850 network processors.
3. Supports Time-Based and Random-Based sampling



4. Supports Monitor and Threshold macros

Assembler

1. Introduced a new naming convention for import variables, which allows the preprocessor to determine whether or not a token is an import variable. The new convention is to place an “i\$” prefix on the name. This is used by the new preprocessor function `isimport(token)` which returns true if the given token begins with the “i\$” prefix. The prefix is stripped prior to writing the .list file, so only microcode needs to be modified.
2. Added new level 4 warning (#5151), which warns when a register inside a nested `begin/end` block hides a register in the enclosing block. This is useful in catching the problem where a macro parameter matches a register declared inside the macro. Note: the warning level can be raised via the `#pragma warning` directive.
3. Added additional “`#pragma optimize`” specifiers, which allow finer control over execution speed optimizations. Please see the Programmer’s Reference Manual for additional information.
4. Increased the maximum number of `begin/end` blocks to 99999.
5. Added “`-lr`” option which produces a “.lri” file. This contains information helping to identify where register congestion is the greatest, particularly in the case where register allocation fails.

Microengine C Compiler

- If critical paths specified with the `__critical_path()` directive overlap, they can be prioritized with an integer. See Section 3.10.1 in the *Intel® IXP2400/IXP2800 Network Processor Microengine C Language Support Reference Manual* for details.
- The `#pragma nounroll` and `#pragma unroll()` directives have been added to allow fine tuning of loop unrolling. See Section 3.13 in the *Intel® IXP2400/IXP2800 Network Processor Microengine C Language Support Reference Manual* for details.
- The `-Qmapvr` switch will now print out a mapping of physical registers to program variables in addition to the information previously printed.
- The `__no_swap_begin()` and `__no_swap_end()` intrinsics can be used to create a section of code where the compiler will not create any instructions which incur a context swap, or move any code into the region that will incur a context swap. See Section 3.12 in *Intel® IXP2400/IXP2800 Network Processor Microengine C Language Support Reference Manual* for details.
- Local variables of inlined functions can now be viewed in the debugger.
- The compiler will now allow generate `T_INDEX` accesses for variable index expressions into transfer register arrays. See Section 3.2.1.2 in the *Intel® IXP2400/IXP2800 Network Processor Microengine C Language Support Reference Manual* for details.
- The compiler now supports the `__crc_none` keyword in intrinsics and inline assembly. This is used to use the byte-swapping capability of the `crc[]` instruction without computing a CRC.
- The compiler supports user-guided optimization of `switch()` expressions. See Section 3.11 in the *Intel® IXP2400/IXP2800 Network Processor Microengine C Language Support Reference Manual* for details.
- The compiler will now generate autoincrement and autodecrement addressing for local memory accesses in loops whose addresses vary by a constant amount. This optimization can be disabled with the switch “`-Qlm_unsafe_addr`”. Please see Section 6.2.6 in the *Intel® IXP2400/IXP2800 Network Processor Microengine C Language Support Reference Manual* for more details.
- A new switch, `-Qperfinfo=0x800`, has been added to show the maximum physical register pressure. An example dump looks like

```
Maximum live physical gprs (52) at myfile.c 245
Maximum live physical gprs (52) at myfile.c 203
Maximum live physical gprs (52) at myfile.c 196
```



Maximum live physical gprs (52) at myfile.c 192

Maximum number of live physical gprs before spilling is printed, including the file name and line number of the high register pressure points.

- Local variables of functions which are inlined will no longer affect register allocation outside the scope of the inlined function.
- The compiler will more effectively use registers and local memory. Spilling and local memory pointer writes have been much reduced.
- Defer[] slot filling has been dramatically improved.
- The compiler version information (displayed by executing “uccl.exe” in me_tools/bin from the command line) appears as follows:

```
Intel(r) Microengine C Compiler
  Version:      3.5
  Release:     1.3
  Build Number: 8.0.13.0 (v1.3)
  Build Date:  Sep  8 2003 01:39:40
Copyright (C) 1985-2003 Intel Corporation.  All rights reserved.
```

“Version” refers to the SDK release that the compiler is targeted towards.

“Release” is the compiler major version number. Changes in the “Release” version usually indicate major changes such as feature set additions.

“Build number” identifies the specific compiler version and is unique for all compiler releases. Higher build numbers always indicate a later compiler.

- The compiler’s assembly output will contain the older “reflect[]” instruction keyword instead of the newer “cap[reflect...]” keyword. Both instructions can be used in inline assembly, and the correct binary opcodes will be generated in either case.
- Remote registers cannot be reflected to with offsets, for example:

```
__declspec(remote, sram_write_reg) a[5];
reflect_write(data, ME, &(a[1]).... );    // Only "&a" can be used; not
                                           // &(a[1]).
```

Separate register variables must be used to perform operations such as the above.

- The compiler now supports a new directive, which is used as follows:

```
#pragma comment(linker ".%linkerdirective arg1 arg2")
```

This directive will emit “.%linkerdirective arg1 arg2” directly at the top of the .list file. It can be used, for example, to emit the .%image_name "imagename" directive which tags a .uof binary with a label.

Linker

- Added the -l switch to disambiguate between microengine number and *.list file path. In earlier versions, if a *.list file name started with a number, that number would be parsed as a microengine number rather than as part of the *.list file name. Using the -l switch permits *.list file names that begin with a number.

Loader

- Increased the maximum number of IVD values to 1024.
- Fixed a bug that prevented the loading of the debugInfo section of the UOF in Linux kernel-mode.



Intel XScale® Core Target Libraries

- The HAL physical address mapping has changed from a constant; therefore, `simIo_Init` then `halMe_Init` must be called prior to using any of the HAL macros, and your application/foreign-model linked with the `halMev2`, `utils.lib`, `ossl.lib`, and `simio_lib.lib` libraries. If you are linking with `loader_dll.lib`, then the `simio_lib.lib` must precede it. Also, your foreign-model code must be compiled as “**Multithreaded DLL**” (see Visual Studio setting “Project Settings->C/C++->Category:Code Generation, Use run-time library”) in order to link properly.

Dataplane Library Support

- New microcode macros added relating to the ‘Relocatable DataStructure’.
- New microcode macros and microengine C APIs added for CAM sharing.
Example programs are provided to demonstrate the usage of these new enhancements.
- The data structure for mailbox in `ixp_lib.h` has changed as follows:

```
typedef struct
{
    union{
        void *addr;
        struct mbox_xfer
        {
            uint32_t THD_ID;
            volatile __declspec(remote sram_read_reg) uint32_t *rem_reg;
        }mbox_xfer;
    };
}mbox_t;
```

NOTE: All crypto related dataplane library code has been moved to the Intel® IXP2850 Simulation Environment CD-ROM.

OSSL Library Support

The API `ix_ossl_thread_create()` has a new parameter `const char* name`: name to be assigned to the newly created thread.

Firmware (Tools CD # 3)

The Firmware source/binary is now released as the Tools CD # 3 for both the Intel® IXDP2400 Advanced Development Platform and the Intel® IXDP2800 Advanced Development Platform. Please note that the Tools CD # 1 should be first installed, before installing Firmware.



Major Bug Fixes

IXP2800 and IXP2400 Transactors

<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
2298	If a script file calls another script file "doesnotexist.ind;" which does not exist crashes. The call to the missing script ends with ';'.	Fixed
5899	Triple port enable/device select 2 generates different bus output	Fixed
6075	Transactor crash using cap_csr_wr with Reference History ON	Fixed
6121	VMOD Array Optimization Issue	Fixed
6122	VMOD 7x Memory size issue	Fixed
6194	TCAM cannot write to QDR_Q_H, QDR_CIN pins	Fixed
6209	Cannot Write to several MSF CSRs in the IXP2800 Transactor	Fixed
6235	Reference History problem with DRAM-RBUF/TBUF on IXP2800	Fixed
6315	Strings of a length longer than 200 characters will crash the transactor.	Fixed
6660	Workbench/Transactor Crash: There is an unsupported format string syntax in the transactor. The %nnX specifier in a format string is not supported (i.e. %4d).	Fixed
6803	Transactor/Workbench may hang when starting a debugging session with a foreign model. This is due to un-released resources held by the foreign model during previous sessions.	Fixed
6957	Transactor hangs with certain foreign model configuration when the following steps are executed: 1) Open project 2) Start (F12) 3) Run (F5) 4) Stop (CTL-F12) after about 2000 cycles 5) Rebuild (ALT-F7) 6) Start (F12)	Fixed
6958	There exists an issue with access to QDR pins after a reset.	Fixed
6993	Transactor problem with 2800B0, 16MEs, Reference History selected on WinXP	Fixed

Developers Workbench

<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
5330	When a breakpoint is set on a instruction that lies within a nested macro the line number reported in the breakpoint-hit dialog is different that the line number in the status bar.	Fixed
6367	"File->Save As" adds extra extension to file name being saved.	Fixed
6361	Workbench allows File->Save to be performed even if the file has not been modified.	Fixed
5031	Go to Line will crash the application if the user attempts to go to line zero.	Fixed
3882	When debugging MicroC code using the source view in a thread window, if you try to set a breakpoint on a line that does not generate code, e.g., a comment line, the Workbench does nothing.	Fixed
6089	Data watches of C variables that are signals should not be allowed.	Fixed
6412	History values for the ctx_enables CSR are not saved correctly if a software breakpoint, ctx_arb[bpt], instruction is executed.	Fixed
IXA00005752	Breakpoint is not reported in hardware mode Debugging.	Fixed
IXA00006890	Add Watch PCI CSR in hardware mode displays error message	Fixed
IXA00024234	Clicking on filter by packet while the GO is in progress, crashes Workbench.	Fixed



Assembler

<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
5298	You can't consume half of a doubled signal using <code>ctx_arb[]</code> .	Fixed
6033	<p>The <code>isnum()</code> preprocessor function does not treat import variables as numeric constants. This is not a defect per se, but may result in unexpected behavior, in particular when import variables are used with standard library macros such as <code>move</code>. For example:</p> <pre>move [gpr, import_var]</pre> <p>expands to:</p> <pre>alu[gpr, --, b, import_var]</pre> <p>because <code>isnum()</code> returns false for an import variable. This would not be the desired result if the import variable is larger than 8 bits.</p> <p>The workaround in this case is to directly use the <code>immed32</code> macro, for example:</p> <pre>Immed32[gpr, import_var]</pre>	Fixed (added new function <code>isimport()</code>)
6273	<p>A reference to a local aggregate neighbor destination register will incorrectly find an old-style aggregate remote register with the same base name. When writing a neighbor destination register, the assembler will preferentially find the remote register. This causes a problem when the local register is an aggregate register and the remote register is an old-style aggregate with the same base name, for example:</p> <pre>.reg remote n\$name0 .reg n\$name[4] immed[n\$name[0], 0] ; DEFECT -- references remote register n\$name0 immed[n\$name[2], 2] ; assembler correctly references local register</pre> <p>In practice this defect is not an issue because:</p> <ol style="list-style-type: none"> 1. local and remote registers cannot be accessed simultaneously, so if there is a remote neighbor register it is in all likelihood the intended target 2. The assembler will report a warning that it matched the "old-style" aggregate register in backwards compatibility mode. <p>The workaround is to rename the local neighbor register.</p>	Fixed
6610	<p>The assembler may generate an incorrect optimization if a <code>.begin</code> or <code>.end</code> directive follows a conditional branch or its deferred uwords (if it has a <code>defer</code> token) and the directive is in a <code>local_csr_wr</code> shadow. The assembler may incorrectly move or remove instructions, causing a later use of the local <code>csr</code> which is being written to fall within the shadow of the <code>local_csr_wr</code>. For example, in the following code, the <code>NOP</code> is incorrectly removed</p> <pre>local_csr_wr [active_lm_addr_0, 0] beq[label#], defer[1] immed[--,0] .begin .reg value nop ; the optimizer will incorrectly remove this NOP alu[value,--,b,*1\$index0]; references old value if NOP is removed alu[--,--,b,value] .end</pre> <p>The workaround is to use the <code>#pragma optimize</code> directive to disable optimizations for the block consisting of the <code>local_csr_wr</code> and the three instructions that follow, for example:</p> <pre>#pragma optimize ("d", off) local_csr_wr [active_lm_addr_0, 0] beq[label#], defer[1] immed[--,0] .begin .reg value nop ; the optimizer will incorrectly remove this NOP #pragma optimize ("d", on) alu[value,--,b,*1\$index0] ; references old value if NOP is removed alu[--,--,b,value] .end</pre>	Fixed
6669	For certain instructions, cryptic error messages are reported if an immediate operand exceeds the	Fixed



<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
	<p>maximum value. For example:</p> <pre>local_csr_wr[active_lm_addr_0, -1]</pre> <p>generates the errors:</p> <pre>(3365) : ERROR: Specified unsigned value -1 does not fit in "abs_immed_datab_8_bit" field. (3365) : ERROR: "-1" is not a recognized value for field "abs_immed_datab_8_bit" in macro "local_csr_wr_i".</pre> <p>The number in parenthesis corresponds to the line number in the intermediate file listfilename.uci.</p>	
6704	<p>An internal error may be generated if the optimizer is enabled and a multi-step instruction such as <code>mul_step</code> or <code>pop_count</code> uses an uninitialized source register. For example:</p> <pre>.reg uninitialized dest mul_step[uninitialized,3], 24x8_start mul_step[uninitialized,3], 24x8_step1 mul_step[dest,--], 24x8_last</pre> <p>will produce an error if the optimizer is enabled. The workaround is to fix the "used before being set" (#4700) warnings which are reported by the assembler.</p>	Fixed

Microengine C Compiler

<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
5776	The compiler was producing incorrect code for the preincrement operator when applied to shared memory variables. This has been fixed.	Fixed
5790	The compiler was not honoring the latency requirement between <code>ctx_arb[--]</code> and <code>local_csr_wr[active_ctx_wakeup_events,..]</code> .	Fixed
5870	Under Workbench, compiler messages were being sent to both standard error and standard input, causing their order to be "scrambled".	Fixed
6863	The compiler was sometimes moving shared local memory variables out of loops if the variable was not written to inside the loop.	Fixed
19501	Bugs relating to the incorrect setting of the local memory pointer index registers were identified and fixed.	Fixed



Known Problems and Workarounds

'Disposition' field can be one of the following:

<i>Blank or Under Investigation</i>	The bug is being investigated.
<i>Root Cause Identified</i>	The root cause for the bug is identified.
<i>Will be Fixed in Release-N</i>	Based on effort estimate, the release the fix will be in.
<i>Workaround Available</i>	The workaround is described in the <i>Release Note</i> column.

IXP2800 and IXP2400 Transactors

<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
4324	Goto /silent is rejected by transactor.	
4325	Goto_addr /silent is not silent.	
	Known issue with using the XACT_get_handle function to obtain a handle to an entire array state. Workaround is to use an index of -1 (instead of -2).	Workaround available
IXA00024340	Reference history not displayed correctly in IXP2800 B0 when using sram atomic commands with the no_pull option. Workaround is to disable reference history	Workaround available
IXA00024356	Two csr issues on ixp2800B0 and ixp2850B0 First issue is writing to ACTIVE_LM_ADDR_0, 1. A write to these csrs through the workbench or SimWrite will not correctly store the data. Second issue is writing the NN_MODE bit (20) of the CTX_ENABLES csr. A write to this csr through the workbench or simWrite will appear to have been stored correctly, but will not have changed the NN_MODE. Workaround for both issues is to use sim_write_special(chip_name, 0x0, addr, value), where address is the Intel XScale® core address of the csr. This function is defined within the init_fuctions_IXP2800.ind, which is loaded by the workbench, and will need to be loaded during console runs.	Workaround available
Vmod - 477	In script files, parsing of some printf and fprintf statements can lead to a unrecoverable error in the transactor (vmod) due to incorrect memory allocation. Workaround - In script files, avoid using <i>printf</i> or <i>fprintf</i> format specifier strings that contain any of the following characters - "-", "+", "'", "#" ... where ' ' represents a space character - in the optional format specifier <i>flag</i> field. The optional format specifier <i>flag</i> field immediately follows the "%" character that initiates a format specifier, and immediately precedes the optional format specifier <i>width</i> field. Avoid using format specifier <i>type-prefix</i> options - "h, l64, l, L". The optional <i>type-prefix</i> format specifier field immediately precedes the mandatory format specifier <i>type</i> field, the last field of each format specifier. Note: the C-interpreter only supports 32-bit integers, so customers should not use integer format specifiers <i>type</i> fields that expect integer sizes other than 32 bits.	Wordaround available – fixed in next release

Developers Workbench

<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
2661	If a simulation breakpoint is set from the command line or from within a script, the Workbench doesn't display the breakpoint marker in the thread window.	Root Cause Identified
3678	Under some conditions, the Workbench controls and windows turn white and are notrefreshed when you start or stop debugging.	Root Cause Identified
4336	If an assembler source file has a multiline C-style comment, i.e., bracketed by /* and */, and you delete a line above that comment, sometimes the syntax coloring for the comment is corrupted.	Root Cause Identified
5181	If you have break on change for all transfer order registers on both the read side and on the write side, and then you remove all the break on changes from the read side and	Root Cause



	<p>stop and start debugging, when you get back, all the break on changes are missing from the write side as well.</p> <p>If you have break on change for all transfer order registers on both the read and the write side, and then you remove the break on change from a single register on the read side and then stop and start debugging, the break on change is missing from the write side register as well.</p> <p>If you there is no break on change on the same register in either the read or the write side, and then you add a break on change on the read side and stop and start debugging, when you get back, the write side will also have a break on change.</p>	Identified
6668	When a register, local memory, cam, or csr data watch value is changed by the user through the data watch debugger view, the new value does not appear in the history	Root Cause Identified
6038	Ctrl-C in a document window copies to the clipboard even if nothing is selected.	Root Cause Identified
	Datatips and data watching for variables with the same name within nested scope on same source line don't work correctly.	Root Cause Identified
	A ";" is not treated as comment delimiter in inline assembler code within a C source file.	Root Cause Identified
	<p>For debug-only projects, the location of the list and source files cannot be changed such that files that were in the same folder when the uof file was created are in different folders.</p> <p>Also, if the list files have been moved, then in order to successfully run a debug-only project as a Workbench batch job you have to first run the Workbench and Start Debugging. This</p>	Workaround Available
5074	In some situations, the Workbench won't respond to the mouse scroll button.	Under investigation
IXA00018366	Double data watches and memory watches after reset. Possible crash if a duplicate watch is deleted.	Root Cause identified. Fixed in next release.
	For SPHY devices which have port numbers that don't start at zero or are non-contiguous (e.g, 0, 2, 3), if the user disables a port, the WB passes the incorrect port number to the MAC model in the phy_enable_port() console function. This causes the MAC model to report an error in a message box and in the command line output window during debug startup. The affected SPHY devices would be those with bus modes of 2x8, 3x8, 1x16, 2x16 and 1x8_1x16.	Root Cause identified. Fixed in next release.
	The bus mode for x16MPHY4 and x16MPHY32 devices was changed when the Westport support was added. As a result, older project files will be read in with these devices having the incorrect bus mode. The effect of this is that the user won't be able to connect a second (SPHY) device to the MSF if this PMHY device is connect as the first device.	Root Cause identified. Fixed in next release.
IXA18469	Due to an issue in the debug libraries, if an SRAM channel is populated with a TCAM followed by SRAM, then the SRAM is not visible in the workbench.	Root Cause identified. Fixed in next release.

Hardware Performance Monitor Limitations

<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
	There are occasional problems with selecting events and/or creating new macros. A work-around is to close the PMU GUI (i.e. return to the Workbench) and re-open the PMU GUI.	
	If you select a threshold event, then try to select a different event, you can only select the new event if it occurs earlier in the list of events. Workaround is by canceling and reopening the dialog box.	
	Editing of macros is not supported. The workaround is to delete that macro and create a new one with the new parameters.	
	PMU support is only provided for targets running Vxworks*. If the PMU GUI is invoked when the target is running a different OS (e.g. Linux), the results are unpredictable.	
	The output file type ".tdh" is not supported and does not appear as a selection.	



Assembler

<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
	<p>XFER_ORDER Issues</p> <p>This is not a bug or a limitation, but it may be an area of confusion.</p> <p>All registers that are related via <code>.xfer_order</code> must have the same scope. So, for example, the following would be invalid:</p> <pre>.begin .reg \$x1 \$x2 .begin .reg \$x3 \$x4 .xfer_order \$x1 \$x2 \$x3 \$x4 // invalid</pre> <p>Similarly, the following would also be invalid:</p> <pre>.reg \$x1 .reg visible \$x2 .xfer_order \$x1 \$x2 // invalid</pre> <p>This is because \$x1 is declared with a module scope, and \$x2 is declared with a global scope (since visible implies global).</p>	
	<p>The assembler does not generate any error message when include path within micro-code file is too long.</p>	
	<p>If a set signal is set again prior to being consumed, the assembler will treat the first I/O operation as never having completed, for example:</p> <pre>.sig s sram[read, \$x, raddr,0, 1], sig_done[s] nop nop sram[read, \$x, raddr,0, 1], sig_done[s] ctx_arb[s]</pre>	
	<p>Comments inside a "&remote" expression are not displayed in the list file. For example:</p> <pre>immed[reg, &remote(rsig, ; this comment is not displayed 0)]</pre>	No fix planned
6533	<p>The constant expression 0xFFFFFFFF is treated as a sign extended value.</p>	
6504	<p>A confusing error message is displayed when a ".local_mem" name and a ".local" directive use the same name. For example:</p> <pre>.local_mem name scratch 10 .local name</pre>	
21595	<p>The "#pragma warning" directive has no effect on warnings generated by the preprocessor.</p>	No fix planned
21228	<p>The assembler inconsistently handles negative operand values for the immed instruction. For certain unresolved expressions, the linker will generate an error saying that the value is too big to fit in the given field. The work-around is to use an expression to mask the resulting value so that it fits in the field.</p>	Workaround Available No fix planned
6436	<p>When using the <code>ind_targets</code> token, the assembler is not asserting that the indirect targets are in 8-ctx mode.</p>	
6434	<p>"<code>immed [register1, register2]</code>" gives a confusing error message. "register2" should be a numeric constant, but the assembler will report "attempt to use undeclared register". If the register is declared, the assembler will then report the correct error.</p>	
6393	<p>If both the input and output files have the same name but differ by path, the assembler will nevertheless report an error that the two files are the same.</p>	No fix planned
6147	<p>Hex indices are not accepted for aggregates within constant expressions, e.g.:</p> <pre>immed [--, (&\$x[0x0])]</pre>	
5741	<p>When assembling from a previously generated <code>.ucp</code> file (via the "-N" command line option), the assembler will not generate warnings for import variables which do not start with the 'i\$' prefix and were used in the <code>isimport()</code> preprocessor function.</p>	
5912	<p>Next neighbor aggregates are not currently supported. For example, the following code will result in a linker error:</p>	



<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
	<pre>.reg remote n\$agg[4] immed[n\$agg[0x0], 0]</pre>	
5972	<p>Under certain circumstances, the optimizer can move instructions in such a way that the instruction sequence in the thread-view window (and list file) appears incorrect. For example, if the initial instruction sequence was:</p> <pre>alu[y,--,b,x] alu[x,--,~b,x]</pre> <p>so that Y gets the value of X, then X gets inverted; then sometimes the optimizer can rearrange the code so that it appears:</p> <pre>alu[x,--,~b,x] alu[y,--,b,x]</pre> <p>It now appears that Y gets the inverted value of X, but due to the way that the registers are allocated, Y actually gets the previous (unmodified) value of X correctly. However, this is confusing to people looking at the output of the assembler. It also causes problems for the workbench, where just before the second statement is executed, it reports the modified value of X in its display windows.</p>	
	<p>Instructions <code>br_bclr</code>, <code>br_bset</code>, <code>br=byte</code>, <code>br!=byte</code> do not support import variables for the bit or byte selection operands. One must be particularly careful in the case of the <code>br_bclr</code> and <code>br_bset</code> instructions, because no error will be reported; the assembler will silently use bit 0. In the case of <code>br=byte</code> and <code>br!=byte</code>, an error will be reported. For example:</p> <pre>.import_var bit_position br_bset[gpr, bit_position, label#]</pre> <pre>.import_var byte_no br=byte[gpr, byte_no, immed_value, label#]</pre> <p>The workaround for this problem is to use an alu instruction and temporary register, for example:</p> <pre>.reg tmp alu[tmp,--,b,gpr,<<((31-bit_position))] br_bset[tmp, 31, label#]</pre> <pre>alu[tmp,--,b,gpr,<<((3-byte_no)*8)] br=byte[tmp, 3, value, label#]</pre>	

Microengine C Compiler

<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
6010	The <code>xscale_int_[ab]</code> CSRs are not yet supported in inline assembly. They can be accessed with the intrinsic functions or referred to in inline assembly by their previous names, <code>thread_interrupt_[ab]</code> .	Root cause identified

Linker

<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
3397	Fixup-values are silently truncated whenever they exceed the assigned field.	

Loader

<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
3397	Fixup-values are silently truncated whenever they exceed the assigned field.	

Debug Library

<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
7085	In the debug library the dram CSRs were not mapped. If you select a DRAM memory CSR in data watch, the Workbench will display "Bad argument error Bad argument error returned from debug"	Fixed



<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
	library when trying to read CSR"	
18469	<p>The HAL exposes a number of pointers/addresses, defined in halMmap.h. Some of these, relating to SRAM addresses, are defined in an inconsistent manner, which may be confusing to users. In particular, there are (where # is the channel number: 0-3):</p> <pre>extern uint64 Hal_sram_ch#_virtAddr; /* SRAM channel # */ extern uint64 Hal_sram_ch#_bit_set_virtAddr; /* SRAM channel # bit set/test */ extern uint64 Hal_sram_ch#_bit_clr_virtAddr; /* SRAM channel # bit clear/test */ extern uint64 Hal_sram_ch#_bit_add_virtAddr; /* SRAM channel # bit add/test */</pre> <p>The first of these point to the beginning of the channel, whereas the rest point to the beginning of the SRAM in the channel. If the channel contains only SRAM, then these all consistently point to the start of the SRAM. But if the channel consists of TCAM followed by SRAM, then the first will point to the start to the TCAM (requiring an offset to get to the SRAM) and the rest will not.</p> <p>For example, if virtual==physical, and if SRAM channel 3 (which starts at 0xB0000000) consists of 0x02000000 bytes of TCAM followed by SRAM, then Hal_sram_ch#_virtAddr == 0xB0000000 but Hal_sram_ch#_bit_set_virtAddr == 0xB6000000. That is, the bit_set address is the start of the channel's memory range (0xB4000000) plus the size of the TCAM (0x02000000).</p> <p>The easiest way to use these in a consistent manner is to make a local pointer which points to the start of the R/W SRAM area (in the above example 0xB2000000). To do this, one should use the function halMe_GetSysMemInfo(). Refer to the header file halMev2Api.h for more information. Once this function is called, then the start of the SRAM portion of the SRAM channel can be computed as:</p> <pre>Hal_SysMemInfo_T sysMemInfo; halMe_GetSysMemInfo(&sysMemInfo); my_Hal_sram_ch#_virtAddr = Hal_sram_ch#_virtAddr + sysMemInfo.sramChan[#].sramOffset;</pre> <p>This will make my_Hal_sram_ch#_virtAddr consistent with Hal_sram_ch#_bit_set_virtAddr etc. Note that if there is no TCAM present, then sysMemInfo.sramChan[#].sramOffset will be zero, and the virtual address will be unchanged.</p>	

Spi4/BFM

<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
	When the receive flow control is turned on, and the calendar length is programmed to be larger than the number of ports in MAC device, the spi4_bfm.dll will corrupt the internal memory structure. The consequence of this problem will fail the transactor to execute the sim_reset.	This problem is identified and fixed on spi4_bfm version 44

Installation

<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
	If IXA SDK 3.51 is installed, then installing IXA SDK 3.1. The portmapper may not be started automatically. Go to Start\Settings\control Pannel\Administrative Tools\Component Services\Services (Local) and start the IXP2000 PortMapper if it's not started.	
	Running SDK3.0 and SDK3.51 installation simultaneously is not recommended.	
24320	When uninstall IXA SDK Tools CD on the XP system. Lock file Detected Window pops up due to portmapper is running. The message says "An option you selected requires that files be installed to or uninstalled from your system, or both. A locked file, C:\Program Files\IXA_SDK_Portmapper\portmapper.exe, was found while performing the needed file operations. To leve this file as it is on your system, click the Ignore button; to retry the file operation, click Retry; or to perform the operation when your system is rebooted, click Reboot."	



<i>Id</i>	<i>Release Note</i>	<i>Disposition</i>
	Click the Ignore button, the uninstallation will continue and uninstall the sdk with no error.	
24482	Occasionally the IXP2000 portmapper service does not get installed on some Windows XP systems after the IXA SDK Tools CD is installed. To install the IXP2000 Portmapper manually after the IXA SDK 3.51 Tools CD is installed. <ol style="list-style-type: none">1. Launch an MS-DOS shell.2. Change the directory to C:\Program Files\IXA_SDK_Portmapper3. C:\Program Files\IXA_SDK_Portmapper>portmapper.exe -install	
18455	If %SystemRoot% or other variables are in the system path, they may stop being expanded after IXA SDK is installed and rebooted. This may cause the path to not work correctly. There are 2 workarounds: <ol style="list-style-type: none">1. change the %SystemRoot% to C:\WINNT or <ol style="list-style-type: none">2. Open the Control Panel -> System -> Advanced -> Environment Variables and edit the system path. Remove the C:\IXA_SDK_3.5\me_tools\bin, click OK. This will force the Path type from REG_SZ back to REG_EXPAND_SZ, then add the C:\IXA_SDK_3.5\me_tools\bin again.	

Compatibility Notes

Linux Emulation Program

You can use a Linux emulation program, for example VMware* Workstation
http://www.vmware.com/vmwarestore/newstore/wkst_eval_login.jsp

to develop in a Linux environment on a Windows 2000* host. After the emulation program is installed, follow the instructions in the *Intel® Internet Exchange Architecture Software Development Kit (IXA SDK) 3.51 Tools Installation Guide* for installing and configuring the tools in a Linux environment.

NOTE: The Linux emulation program/Windows 2000 host configuration should be used for testing purposes only. It is not officially supported and has not been tested.

Linux Host Environment

Red Hat* 7.3 Linux* is supported as the development host OS for IXA SDK 3.51. However, Red Hat does not officially support this version of the OS. Red Hat* 7.3 Linux* can still be obtained through PC Mall (<http://www.pcmall.com>).

“Look-alike” versions of Linux are also available from other sites such as <http://www.cheapbytes.com>. However, these versions of Linux OS have not been tested with IXA SDK 3.51 and Intel does not give any guarantees regarding interoperability with these OSs.

Documentation Issues

None for this release of the product.



Common User Problems

1	<p>Unexpected return status (0x100de) from udebug library when trying to Pause ...</p> <p>Description: When trying to pause the microengines, or when hitting a breakpoint, the Workbench reports an error "Unexpected return status (0x100de) from udebug library when trying to Pause MicroEngines."</p> <p>Cause: This should only occur when running on hardware using the WTX interface and when there is a thread running in a loop with no context arbing instructions. In this case, the thread cannot be successfully paused. The debug libraries wait 5 seconds before concluding that the thread will not stop. However, the default timeout on the target server is 3 seconds (the installation guide specifies that the timeout should be set to 10 seconds). If the timeout is left at 3 seconds, then the target server times out before the debug libraries return the proper "failed to pause" return code. This results in a WTX error value of 0x100de.</p> <p>Fix: Change the timeout to 10 seconds.</p>
2	<p>Installer has detected the Intel® IXA SDK 3.51 Tools on this system ...</p> <p>Description: When installing a new version of SDK 3.51, the installer gives the warning "install has detected Intel® IXA SDK 3.51 Tools on this system" and would not continue.</p> <p>Cause: This is because the Devworkbench.exe or some other files were running during a previous uninstall and the system was not rebooted as instructed by the uninstallation.</p> <p>Work around: Delete the Intel® IXA SDK 3.5 registry key. Go to Start ->Run -> regedit and find the Intel® IXA SDK 3.5 key under HKEY_LOCAL_MACHINE\SOFTWARE\INTEL.</p>

SDK/Firmware Matrix

IXDP2400

The following table indicates the relationship between the Firmware version and the SDK releases. Please make sure the BSP and Boot Monitor versions are in sync. The build dates will be displayed when the IXDP2400 system is booted.

IXDP2400 System matrix

SDK PR #	Firmware ver #	VxWorks BSP build date		Boot Monitor build date
		Bootrom	VxWorks	
SDK PR6- FCS	7a	Feb 27, 2003	Feb 24, 2003	Jan 31, 2003
SDK Tools 3.1 Pre-release 1 and 2	7a	Feb 27, 2003	Feb 24, 2003	Jan 31, 2003
SDK Tools 3.5 Pre-release 1	8a	July 21, 2003	July 21, 2003	July 14, 2003
SDK Tools 3.5 Pre-release 2	9	Nov 05, 2003	Nov 05, 2003	Oct 07, 2003
SDK Tools 3.51 (3.5 re-release)	9a	Mar 12, 2004	Mar 12, 2004	Oct 07, 2003



Firmware Release Notes

VxWorks

This release of SDK supports WindRiver Tornado* 2.2.1. There are two ways to move to Tornado 2.2.

A. Upgrade using WindRiver's CP1 patch:

1. Install TOR2.2 TDK-14622-ZC-01 <ignore this step if you already have T2.2>
2. Install cp1 arch patch t22-cp1-strongarm_xscale.tar
< https://www.windriver.com/cgi-bin/windsurf/downloads/view_binary.cgi?binaryid=909 >
3. Install cp1 drivers patch t22-cp1-drivers.tar
< https://www.windriver.com/cgi-bin/windsurf/downloads/view_binary.cgi?binaryid=909 >
4. Install X-bit patch from windsurf <SPR 89608>
5. Install latest BSP
< <https://www.windriver.com/cgi-bin/windsurf/bsp/infoBSP.cgi?id=366> >
6. Modify BSP's config.h
 - change console baud rate, if needed, as appropriate.
7. Rebuild BSP and test.

B. Upgrade using the Tornado 2.2.1 CD:

Follow Steps 4, 5, 6 and 7 noted in step (A) above.

The complete description of the Tornado 2.2.1 CD and CP1 can be found at:

http://www.windriver.com/support/resources/tornado22_bulletin.html

Linux

IXDP2400:

An older version of LSP (GA) with MontaVista* MVL 3.0 Professional Edition needs the patch 'ixdp2400-phase8a-patch.diff' (packaged in Tools CD # 3 - Firmware, under Linux), whereas the newer version of LSP does not need this patch. Here is how to find out if a patch needs to be applied or not.

After installing MontaVista* MVL 3.0 Professional Edition (GA), do the following:

```
Bash% rpm -q -a | grep ixdp2400
```

If you get:

```
hhl-arm_xscale_be-lsp-intel-ixdp2400-2.4.18_mv130-mv13.0.0.9
```

```
hhl-cross-arm_xscale_be-lsp-intel-ixdp2400-2.4.18_mv130-mv13.0.0.9
```

it means that you are running the latest version (3.0.0.9). Anything else is an older version and the patch needs to be applied.

Here are the commands to apply the patch:



```
bash% cd /opt/hardhat/devkit/lsp/intel-ixdp2400-arm_xscale_be
bash% cat ixdp2400-phase8a-patch.diff | patch -p0
```

IXDP2401:

NOTE: To use the Intel® IXA SDK 3.51 release with the Intel® IXDP2401platform, you need to install Version 1.2 of the Intel® IXA Software Developers Kit (SDK) Tools 3.51 Firmware and Drivers for the IXDP2401 Advanced Development Platform CD-ROM. Versions 1.0 and 1.1 are compatible with SDK 3.1, not this SDK 3.51 release.

File Changes from the Intel® IXA SDK 3.5 to the Intel® IXA SDK 3.51

OSSL changes for the Linux* kernel implementation

The files and code changes of the OSSL Linux* kernel implementation are as follows:

semaphore.c

The function `__down_timeout()` has been replaced with `_ix_ossl_sem_down_timeout()`. A new logic is used in the implementation, and the new code better follows the IXA SDK coding style and conventions. The logic of the code has been changed accordingly to use the new function.

All the `init_MUTEX_LOCKED(pSemaphore)` calls have been replaced with `sema_init(pSemaphore, x)`

thread.c

The following functions have been replaced to better follow the SDK coding style and conventions.:

`sys_sched_get_priority_max()` with `_ix_get_priority_max()`

`sys_sched_get_priority_max()` with `_ix_get_priority_max()`

`_ix_ossl_setscheduler()` with `_ix_set_ossl_policy_and_priority()`

All the references to `kfree()` have been replaced with `ix_ossl_free()`

The following include headers have been removed:

```
#include <linux/spinlock.h>
```

```
#include <asm/uaccess.h>
```

mutex.c

Due to the fact that the function `__down_timeout()` has been replaced with `_ix_ossl_sem_down_timeout()` in the `semaphore.c`, the references in this file to `__down_timeout()` have been replaced with `_ix_ossl_sem_down_timeout()`. The logic of the code has been changed accordingly to use the new function.

internal_os_types.h

The following Linux kernel header files have been added to the set of include files:

```
#include <linux/spinlock.h>
```

```
#include <asm/uaccess.h>
```

```
#include <asm/smplock.h>
```



The following files contain updated version information and new Workbench icon bitmap information:

IXA_SDK_3.5\Documentation\Tools\InstallationGuideForTools.pdf
IXA_SDK_3.5\me_tools\bin\DevWorkbench.exe
IXA_SDK_3.5\me_tools\bin\DEVWORKBENCH.HLP

The following files re-built to include the changes in OSSL:

IXA_SDK_3.5\me_tools\bin\loader_dll.dll
IXA_SDK_3.5\me_tools\bin\parser.dll
IXA_SDK_3.5\me_tools\bin\platform.dll
IXA_SDK_3.5\me_tools\bin\platune.dll
IXA_SDK_3.5\me_tools\bin\ppplwin.dll
IXA_SDK_3.5\me_tools\bin\spl.dll
IXA_SDK_3.5\me_tools\bin_linux_be\halMeDrv.o
IXA_SDK_3.5\me_tools\bin_linux_be\WBSrvr
IXA_SDK_3.5\me_tools\bin_linux_be\debug\halMeDrv.o
IXA_SDK_3.5\me_tools\bin_linux_be\debug\WBSrvr
IXA_SDK_3.5\me_tools\bin_linux_kernel_be\halMeDrv.o
IXA_SDK_3.5\me_tools\bin_linux_kernel_be\halMev2_lib.o
IXA_SDK_3.5\me_tools\bin_linux_kernel_be\libossl.o
IXA_SDK_3.5\me_tools\bin_linux_kernel_be\debug\halMeDrv.o
IXA_SDK_3.5\me_tools\bin_linux_kernel_be\debug\halMev2_lib.o
IXA_SDK_3.5\me_tools\bin_linux_kernel_be\debug\libossl.o
IXA_SDK_3.5\me_tools\bin_vxw_be\WBSrvr.o
IXA_SDK_3.5\me_tools\bin_vxw_be\debug\WBSrvr.o
IXA_SDK_3.5\me_tools\lib\loader_dll.lib
IXA_SDK_3.5\me_tools\lib\uclo.lib
IXA_SDK_3.5\me_tools\lib\debug\loader_dll.lib
IXA_SDK_3.5\me_tools\lib\debug\uclo.lib
IXA_SDK_3.5\me_tools\lib_linux_be\dbgMe.a
IXA_SDK_3.5\me_tools\lib_linux_be\debug
IXA_SDK_3.5\me_tools\lib_linux_be\halMev2.a
IXA_SDK_3.5\me_tools\lib_linux_be\libossl.a
IXA_SDK_3.5\me_tools\lib_linux_be\osApi.a
IXA_SDK_3.5\me_tools\lib_linux_be\RdDriver.a
IXA_SDK_3.5\me_tools\lib_linux_be\rs_cntl.a
IXA_SDK_3.5\me_tools\lib_linux_be\rs_udebug.a
IXA_SDK_3.5\me_tools\lib_linux_be\uclo.a
IXA_SDK_3.5\me_tools\lib_linux_be\utils.a
IXA_SDK_3.5\me_tools\lib_linux_be\debug\dbgMe.a
IXA_SDK_3.5\me_tools\lib_linux_be\debug\halMev2.a
IXA_SDK_3.5\me_tools\lib_linux_be\debug\libossl.a
IXA_SDK_3.5\me_tools\lib_linux_be\debug\osApi.a
IXA_SDK_3.5\me_tools\lib_linux_be\debug\RdDriver.a
IXA_SDK_3.5\me_tools\lib_linux_be\debug\rs_cntl.a
IXA_SDK_3.5\me_tools\lib_linux_be\debug\rs_udebug.a
IXA_SDK_3.5\me_tools\lib_linux_be\debug\uclo.a
IXA_SDK_3.5\me_tools\lib_linux_be\debug\utils.a
IXA_SDK_3.5\me_tools\lib_linux_kernel_be\utils_kernel.a
IXA_SDK_3.5\me_tools\lib_linux_kernel_be\debug\utils_kernel.a
IXA_SDK_3.5\me_tools\lib_vxw_be\dbgMe.a
IXA_SDK_3.5\me_tools\lib_vxw_be\halMev2.a
IXA_SDK_3.5\me_tools\lib_vxw_be\libossl.a
IXA_SDK_3.5\me_tools\lib_vxw_be\osApi.a



IXA_SDK_3.5\me_tools\lib_vxx_be\RdDriver.a
IXA_SDK_3.5\me_tools\lib_vxx_be\rs_cntl.a
IXA_SDK_3.5\me_tools\lib_vxx_be\rs_udebug.a
IXA_SDK_3.5\me_tools\lib_vxx_be\uclo.a
IXA_SDK_3.5\me_tools\lib_vxx_be\utils.a
IXA_SDK_3.5\me_tools\lib_vxx_be\debug\dbgMe.a
IXA_SDK_3.5\me_tools\lib_vxx_be\debug\halMev2.a
IXA_SDK_3.5\me_tools\lib_vxx_be\debug\libossil.a
IXA_SDK_3.5\me_tools\lib_vxx_be\debug\osApi.a
IXA_SDK_3.5\me_tools\lib_vxx_be\debug\RdDriver.a
IXA_SDK_3.5\me_tools\lib_vxx_be\debug\rs_cntl.a
IXA_SDK_3.5\me_tools\lib_vxx_be\debug\rs_udebug.a
IXA_SDK_3.5\me_tools\lib_vxx_be\debug\uclo.a
IXA_SDK_3.5\me_tools\lib_vxx_be\debug\utils.a

The following files removed from the CD kit:

IXA_SDK_3.5\me_tools\XSC_CoreLibs\halMev2\halSysMem.c

The following files contain updated licensing information:

IXA_SDK_3.5\me_tools\bin\connect_chips_IXP2400.ind
IXA_SDK_3.5\me_tools\bin\connect_chips_IXP2800.ind
IXA_SDK_3.5\me_tools\bin\format.dll
IXA_SDK_3.5\me_tools\bin\init_functions_IXP2400.ind
IXA_SDK_3.5\me_tools\bin\init_functions_IXP2800.ind
IXA_SDK_3.5\me_tools\bin\init_IXP2400.ind
IXA_SDK_3.5\me_tools\bin\init_IXP2800.ind
IXA_SDK_3.5\me_tools\bin\IXP2400_B0.dll
IXA_SDK_3.5\me_tools\bin\IXP2400_B0.exe
IXA_SDK_3.5\me_tools\bin\IXP2800_A1.dll
IXA_SDK_3.5\me_tools\bin\IXP2800_A1.exe
IXA_SDK_3.5\me_tools\bin\ixp2800_a2.dll
IXA_SDK_3.5\me_tools\bin\ixp2800_a2.exe
IXA_SDK_3.5\me_tools\bin\IXP2800_B0.dll
IXA_SDK_3.5\me_tools\bin\IXP2800_B0.exe
IXA_SDK_3.5\me_tools\bin\support_routines.ind
IXA_SDK_3.5\me_tools\bin\support_routines_IXP2400.ind
IXA_SDK_3.5\me_tools\bin\lib\spixp24.dll
IXA_SDK_3.5\me_tools\bin\lib\spixp28.dll
IXA_SDK_3.5\me_tools\include\foreign_model_user.h
IXA_SDK_3.5\me_tools\include\pciconfx.h
IXA_SDK_3.5\me_tools\include\RDinterface.h
IXA_SDK_3.5\me_tools\include\ossil_include\linux_kernel\internal_os_types.h
IXA_SDK_3.5\me_tools\lib\IXP2400_B0.lib
IXA_SDK_3.5\me_tools\lib\IXP2800_A1.lib
IXA_SDK_3.5\me_tools\lib\ixp2800_a2.lib
IXA_SDK_3.5\me_tools\lib\IXP2800_B0.lib
IXA_SDK_3.5\me_tools\lib\debug\IXP2400_B0.lib
IXA_SDK_3.5\me_tools\lib\debug\IXP2800_A1.lib
IXA_SDK_3.5\me_tools\lib\debug\ixp2800_a2.lib
IXA_SDK_3.5\me_tools\lib\debug\IXP2800_B0.lib
IXA_SDK_3.5\me_tools\PMU\shglobal.h
IXA_SDK_3.5\me_tools\PMU\common\SHtypes.h
IXA_SDK_3.5\me_tools\PMU\driver\DRring.h
IXA_SDK_3.5\me_tools\PMU\elist\SHinst.h
IXA_SDK_3.5\me_tools\PMU\platform\PLglobal.h



IXA_SDK_3.5\me_tools\PMU\RemoteDriver\eventlog.c
IXA_SDK_3.5\me_tools\PMU\RemoteDriver\eventlog.h
IXA_SDK_3.5\me_tools\PMU\RemoteDriver\RDcallback.c
IXA_SDK_3.5\me_tools\PMU\RemoteDriver\RDcommon.h
IXA_SDK_3.5\me_tools\PMU\RemoteDriver\RdDriver.mak
IXA_SDK_3.5\me_tools\PMU\RemoteDriver\RDglobal.h
IXA_SDK_3.5\me_tools\PMU\RemoteDriver\RDhss.c
IXA_SDK_3.5\me_tools\PMU\RemoteDriver\RDhss.h
IXA_SDK_3.5\me_tools\PMU\RemoteDriver\RDinstr.c
IXA_SDK_3.5\me_tools\PMU\RemoteDriver\RDinstr.h
IXA_SDK_3.5\me_tools\PMU\RemoteDriver\RDioctl.c
IXA_SDK_3.5\me_tools\PMU\RemoteDriver\RDioctluser.c
IXA_SDK_3.5\me_tools\PMU\RemoteDriver\RDtimer.c
IXA_SDK_3.5\me_tools\PMU\RemoteDriver\RDtimer.h
IXA_SDK_3.5\me_tools\PMU\RemoteDriver\RDutility.c
IXA_SDK_3.5\me_tools\PMU\RemoteDriver\RDutility.h
IXA_SDK_3.5\me_tools\XSC_CoreLibs\dbgMe\dbgMe_ver.h
IXA_SDK_3.5\me_tools\XSC_CoreLibs\halMev2\halMev2_ver.h
IXA_SDK_3.5\me_tools\XSC_CoreLibs\uclo\uclo.c
IXA_SDK_3.5\me_tools\XSC_CoreLibs\uclo\uclo_ver.h
IXA_SDK_3.5\src\EXAMPLES\microc\api_usage\Example3_ipv4_6\dbcst_table_init_ipv4_5.ind
IXA_SDK_3.5\src\EXAMPLES\microc\api_usage\Example3_ipv4_6\ipheaders_castine.ind
IXA_SDK_3.5\src\EXAMPLES\microc\api_usage\Example3_ipv4_6\routes_table_init.ind
IXA_SDK_3.5\src\EXAMPLES\microc\api_usage\Example6_critsec\example6_critsect1.c
IXA_SDK_3.5\src\EXAMPLES\microc\api_usage\Example6_critsec\example6_critsect2.c
IXA_SDK_3.5\src\EXAMPLES\microc\api_usage\Example7_dram_rbuf\example7_rbuf_dram.ind
IXA_SDK_3.5\src\EXAMPLES\microc\byte_align\ether_packet.ind
IXA_SDK_3.5\src\EXAMPLES\microc\byte_align\ether_packet_le.ind
IXA_SDK_3.5\src\EXAMPLES\microcode\byte_align\byte_align_rbuf\byte_align_be_init.ind
IXA_SDK_3.5\src\EXAMPLES\microcode\byte_align\byte_align_rbuf\msf.uc
IXA_SDK_3.5\src\EXAMPLES\microcode\cam_sharing\dl_system.h
IXA_SDK_3.5\src\EXAMPLES\microcode\fold\test_result_ixp2400.txt
IXA_SDK_3.5\src\EXAMPLES\microcode\fold\test_result_ixp2800.txt
IXA_SDK_3.5\src\EXAMPLES\microcode\lpm\rtm.ind
IXA_SDK_3.5\src\EXAMPLES\microcode\lpm\start_init_ixp2400.ind
IXA_SDK_3.5\src\EXAMPLES\microcode\lpm\start_init_ixp2800.ind
IXA_SDK_3.5\src\EXAMPLES\microcode\meter\meter_init.ind
IXA_SDK_3.5\src\EXAMPLES\microcode\meter\meter_init1.ind
IXA_SDK_3.5\src\EXAMPLES\microcode\meter\meter_init2.ind
IXA_SDK_3.5\src\EXAMPLES\microcode\meter\meter_post.ind
IXA_SDK_3.5\src\EXAMPLES\microcode\meter\meter_post1.ind
IXA_SDK_3.5\src\EXAMPLES\microcode\meter\meter_post2.ind
IXA_SDK_3.5\src\EXAMPLES\microcode\scratch_ring\multi_producer\scratchring.h
IXA_SDK_3.5\src\EXAMPLES\microcode\signal\same_me\support_routines.ind
IXA_SDK_3.5\src\library\dataplane_library\microcode\cam_sharing.uc

IXDP2400 Firmware file changes made for this release:

The following files contain updated licensing information:

Diagnostics:

u2h-2media.bat
u2h-emi.bat
u2h-media_sf.bat
u2h-medlb-JF.bat
u2h-mlb.bat



u2h-mphy-lb.bat
u2h-msf.bat
u2h-pl_linerate.bat
u2h-pl_medlb.bat
u2h-pl_syslb.bat
u2h-rf.bat
u2h-sflb.bat
u2h-syslb.bat
u2h-syslb2.bat
u2h-syslb3.bat
u2h-ucode2h.bat
BIC_README.txt
list_extractor.C
dual_media.h
ethernet.h
analog_bd_eg_rx.h
analog_bd_eg_tx.h
analog_bd_in_rx.h
analog_bd_in_tx.h
dual_bd_eg_rx.h
dual_bd_eg_tx.h
dual_bd_in_rx.h
dual_bd_in_tx.h
dual_bw_eg_rx.h
dual_bw_eg_tx.h
dual_bw_in_rx.h
dual_bw_in_tx.h
dual_pl_eg_rx.h
dual_pl_eg_tx.h
dual_pl_in_rx.h
dual_pl_in_tx.h
emi_bd_rx.h
emi_bd_tx.h
emi_bw_rx.h
emi_bw_tx.h
emi_pl_rx.h
emi_pl_tx.h
line_rate_bd_eg_rx.h
line_rate_bd_eg_tx.h
line_rate_bd_in_rx.h
line_rate_bd_in_tx.h
line_rate_bw_eg_rx.h
line_rate_bw_eg_tx.h
line_rate_bw_in_rx.h
line_rate_bw_in_tx.h
line_rate_pl_eg_rx.h
line_rate_pl_eg_tx.h
line_rate_pl_in_rx.h
line_rate_pl_in_tx.h
media_lb_bd_on_sf_iface_rx.h
media_lb_bd_on_sf_iface_tx.h
media_lb_bw_on_sf_iface_rx.h
media_lb_bw_on_sf_iface_tx.h
media_lb_pl_on_sf_iface_rx.h
media_lb_pl_on_sf_iface_tx.h
media_loopback_bd_rx.h



media_loopback_bd_tx.h
media_loopback_bw_atm_rx.h
media_loopback_bw_atm_tx.h
media_loopback_bw_rx.h
media_loopback_bw_tx.h
ME_4k_Control_Store.h
ME_Context_Test.h
ME_Register_Test.h
ME_Timers_Counters_Test.h
mphy_loopback_eg_rx.h
mphy_loopback_eg_tx.h
mphy_loopback_in_rx.h
mphy_loopback_in_tx.h
msf_cbus_rx.h
msf_cbus_tx.h
msf_unicast_rx.h
msf_unicast_tx.h
pl_media_loopback_rx.h
pl_media_loopback_tx.h
pl_media_lpbk_jumboFrame_rx.h
pl_media_lpbk_jumboFrame_tx.h
pl_sys_loopback_eg_rx.h
pl_sys_loopback_eg_tx.h
pl_sys_loopback_in_rx.h
pl_sys_loopback_in_tx.h
rbuf_flush_bd.h
rbuf_flush_bw.h
sf_loopback_rx.h
sf_loopback_tx.h
sys_loopback_bd_2_eg_rx.h
sys_loopback_bd_2_eg_tx.h
sys_loopback_bd_2_in_rx.h
sys_loopback_bd_2_in_tx.h
sys_loopback_bd_3_eg_rx.h
sys_loopback_bd_3_eg_tx.h
sys_loopback_bd_3_in_rx.h
sys_loopback_bd_3_in_tx.h
sys_loopback_bd_eg_rx.h
sys_loopback_bd_eg_tx.h
sys_loopback_bd_in_rx.h
sys_loopback_bd_in_tx.h
sys_loopback_bw_2_eg_rx.h
sys_loopback_bw_2_eg_tx.h
sys_loopback_bw_2_in_rx.h
sys_loopback_bw_2_in_tx.h
sys_loopback_bw_3_eg_rx.h
sys_loopback_bw_3_eg_tx.h
sys_loopback_bw_3_in_rx.h
sys_loopback_bw_3_in_tx.h
sys_loopback_bw_eg_rx.h
sys_loopback_bw_eg_tx.h
sys_loopback_bw_in_rx.h
sys_loopback_bw_in_tx.h
sys_loopback_pl_2_eg_rx.h
sys_loopback_pl_2_eg_tx.h
sys_loopback_pl_2_in_rx.h



sys_loopback_pl_2_in_tx.h
sys_loopback_pl_3_eg_rx.h
sys_loopback_pl_3_eg_tx.h
sys_loopback_pl_3_in_rx.h
sys_loopback_pl_3_in_tx.h
diag.h
dp_proto.h
error_code.h
error_map.h
i2c_diag.h
mac_diag.h
mac_init.h
mac_util.h
packet_pool.h
prototypes.h
reg_api.h
register_map.h
Makefile
error_map.c
mac_diag.c
mac_init.c
mac_util.c
reg_api.c
led_diag.c
packet_pool.c
per_port.c
rmon_diag.c
wmark_diag.c
dual_bd_eg_rx.uc
dual_bd_eg_tx.uc
dual_bd_in_rx.uc
dual_bd_in_tx.uc
dual_bw_eg_rx.uc
dual_bw_eg_tx.uc
dual_bw_in_rx.uc
dual_bw_in_tx.uc
dual_pl_eg_rx.uc
dual_pl_eg_tx.uc
dual_pl_in_rx.uc
dual_pl_in_tx.uc
line_rate_bw.h
line_rate_pl.h
sys_loopback_bd_2.h
sys_loopback_bd_3.h
sys_loopback_bw_2.h
sys_loopback_bw_3.h
sys_loopback_pl_2.h
sys_loopback_pl_3.h
ethernet.c
lpbk_diag.c

VxWorks BSP (sources available from WindRiver*):

Makefile
aduc812.c
aduc812.h
bootconfig.c

**Intel® IXA SDK Tools 3.51
Software Product Release Notes**

C16902-026

3/25/2004



bootinit.c
config.h
confignet.h
flashmem.c
ixdp2400.c
ixdp2400.h
ixdp2400I2C.h
ixdp2400i2c.c
ixdp2400intrctl.c
ixdp2400misc.h
ixdp2400pci.c
ixdp2400pci.h
ixdp2400sio.c
ixdp2400sio.h
ixdp2400timer.c
pciomaplib.c
pciomaplib.h
pciomapshow.c
rominit.s
sysalib.s
sysend.c
syslib.c
sysserial.c
target.nr
usrconfig.c
usrextra.c
usnetwork.c

IXD2448/IXD2412 VxWorks Driver:

moveallobjs.bat
Makefile
os.h

IXD2410 VxWorks Driver:

Makefile_pl
ixf1104ce_driver_api.h
ixf1104ce_driver_api.c
ixf1104ce_internal.h
ixf1104ce_ioctl.h
ixf1104ce_config.c
ixf1104ce_get_ioctl.c
ixf1104ce_set_ioctl.c

IXD2448/IXD2412 Linux Kernel Mode Driver:

ixf6048_irq.c
ixf6048_irq.h
Linux_IXD2448_kernel_Mode_README.txt
Rules.make
ixf6048_ext_lpbk.h
ixf6048_internal.h
ixf6048_ioctl.h
ixf6048_config.c
ixf6048_get_ioctl.c
ixf6048_set_ioctl.c
os.h



ixf6048_linux_driver.h
ixf6048_linux_driver.c

IXD2410 Linux Kernel Mode Driver:

Makefile
ixf1104ce_internal.h
ixf1104ce_ioctl.h
ixf1104ce_get_ioctl.c
ixf1104ce_config.c
ixf1104ce_set_ioctl.c
ixf1104ce_linux_driver.h
ixf1104ce_linux_driver.c
ixf1104ce_linux_interface.c

IXDP2800 Firmware file changes made for this release:

VisionWare Boot Manager/Diagnostics

Updated Licensing information in the following files

ethernet.c

ether_net.h

copy2flash.h

Removed unused source code files IXP2000gpio.h and IXP2000i2c.h

VxWorks* BSP

Updated licensing information for the following files:

sysEnd.c

usrNetwork.c

usrExtra.c

usrConfig.c

ixdp2000Pci.h

flashMem.c

configNet.h

bootInit.c

bootConfig.c

pciIomapShow.c

pciIomapLib.h

ixdp2000Sio.h

sysaLib.s

target.nr

sysSerial.c

sysLib.c

ixdp2000Sio.c

ixdp2000Pci.c

ixdp2000IntrCtl.c

ixdp2000.h

config.h

romInit.s

pciIomapLib.c

Makefile

Ixdp2000Timer.c

Ixdp2000Timer.h

Ixdp2000.c

Ixdp2800.h

Ixdp2000I2c.h

Ixdp2000I2c.c



Ixdp2000misc.h

IXD28192 vxWorks Device Driver:

Updated licensing information in the following files:

ixf18100_pos_b.h
ixf18100_spi4_b.h
moveallobjs.bat
prjObjs.lst

The following file was removed form this release:

envset.bat

IXD2810 vxWorks Device Driver:

Updated licensing information in the two makefiles

IXD28192 Linux Device Driver:

Updated licensing information in the following files:

ixf18100_pos_b.h
ixf18100_spi4_b.h
moveallobjs.bat
prjObjs.lst
linuxkern.h
linux_int.c
movedrvobjjs.bat
test18101.c

IXD2810 Linux Device Driver:

Updated licensing information in the following files:

/ixf1110/bld1110.h

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. 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.

Contact your local sales office to obtain the latest specifications before placing your order. Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication or disclosure is subject to restrictions stated in Intel's Software License Agreement, or in the case of software delivered to the government, in accordance with the software license agreement as defined in FAR 52.227-7013.

Copyright © 2004 Intel Corporation.

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

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