IX-API SDK 3.0 Release Notes

for the Windows NT® operating system

June 2000. Software Release SDK 3.0
Document Revision 4.0
Part Number 6750006

This document as well as the software described in it is furnished under license and may be used or copied only in accordance with the terms of the license. The information in this manual 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 permission of Intel Corporation.

Intel Corporation might have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give any license to these patents.

Copyright © 2000 Intel Corporation. All rights reserved.

Intel and the Intel logo are registered trademarks and Internet Exchange, NetBoost, NCL, and the NetBoost logo are trademarks of Intel Corporation in the United States and other countries.

ARM and StrongARM are trademarks of Advanced RISC Machines, Ltd.
InstallShield is a registered trademark and service mark of InstallShield Software Corporation in the United States and/or other countries.
UNIX is a registered trademark of The Open Group in the US and other countries.
Windows NT is a registered trademark of Microsoft Corporation.

*Other third-party brands and names are the property of their respective owners.

This product includes software developed by parties other than Intel. **See the back page of this document for a list of copyrights and license agreements.**

Intel Corporation
1350 Villa Street
Mountain View, CA 94041-1126
Tel: 650.567.9800
Fax: 650.567.9810
www.intel.com

Contents

SDK 3.0 Release Information 1

Product Name Changes 1

New Features in This Release 2
- ODX NIC Interface 2
- New NCL Constructs 2
- Diagnostic Utility 3
- New Demos 3

Changes in This Release 5
- Plumber and Factories Are No Longer Supported 5
- Microsoft Visual C++ 6.0 Project Is Required 5
- System ACE NBSystemApp Alias Is No Longer Supported 5
- Integer Type Casting Is No Longer Automatic 6

Known Limitations of SDK 3.0 6
- General Limitations 6
  - Only Static Libraries Are Available 6
  - Multiple Policy Accelerators Are Not Supported 6
  - Number of Concurrent ACEs 6
  - MAC Interface Always in Promiscuous Mode 6
  - Stopping and Starting Applications 6
  - Stopping the Resolver 6
  - Stopping and Starting the PA100 Drivers 7
- NCL and ASL Libraries 7
  - NCL Code Size Limitations 7
  - Set Restrictions 7
  - NCL Syntax Errors 7
  - Static Initialization Fails in Action Code 7
  - Compiler Warnings 7
- Action Code Debugger (nbgdb) 8
  - Serial Port Connection Problem 8
  - nbgdb Does Not Support More Than One ACE Sharing Action Code 8
  - Delete Breakpoints Before Detaching the Process 8
  - Use a Local Variable to Bypass Casting Bug in gdb 8
- ODX NIC Interface 8

Directly Connected NICs 8
Monitoring Features Not Supported for Interface C 9
Reported Incidents Fixed in This Release 9
Porting Applications from Previous Releases 9
Porting from the SDK 2.0 Release 9
Documentation for This Release 10
Installation Instructions 10
Documents Formats 10
New Documents 10
Documents Updated 10
Location 11
Errata 11

SDK 3.0 Release Information

This document describes features and behavior of the 3.0 release of the Intel® IX-API SDK, and is a supplement to the Intel documents *Developing Applications Using the IX-API SDK* and the *IX-API SDK Reference.*

NOTE: This release supports *only* the Windows NT® operating system. System requirements are provided with the installation instructions in the document *Installing the IX-API SDK.*

This document contains the following sections:

- "Product Name Changes" on page 1
- "New Features in This Release" on page 2
- "Changes in This Release" on page 5
- "Known Limitations of SDK 3.0" on page 6
- "Reported Incidents Fixed in This Release" on page 9
- "Porting Applications from Previous Releases" on page 9
- "Documentation for This Release" on page 10

Product Name Changes

The following product names have changed for this release:

- Policy Accelerator 100 is the new name for the Policy Engine
- IX-API SDK is the new name for the software developer's kit for the Policy Accelerator 100

Documents and software have been updated to reflect these changes and related changes in terminology. For example, the *PE module* of a *PolicyWare application* is now known as the *accelerator module* of an *IX application*.

New Features in This Release

This release includes new features in the following areas:

- "ODX NIC Interface" on page 2
- "New NCL Constructs" on page 2
- "Diagnostic Utility" on page 3
- "New Demos" on page 3

ODX NIC Interface

For Windows NT, this release supports a direct interface between the Policy Accelerator board and a standard network interface card (NIC) through the PCI bus. To establish this interface, you must customize your NIC driver using the Optimal Data Exchange (ODX) Protocol. The Policy Accelerator and NIC then use the ODX protocol to pass packets back and forth through the PCI bus.

For information on how to customize a NIC driver using the ODX protocol, see *Customizing a NIC Driver Using the ODX Protocol*. A diagnostic utility for testing the NIC interface when you have completed the customization is available as part of the SDK installation at the following location:

SDKinstallpath/diagnostics/odxloop

After you have customized the driver, the NIC interface is available as interface C to the Policy Accelerator. Interface C behaves exactly like the built-in interfaces A and B on the Policy Accelerator boards. You can bind targets to it using the name **nbhwpe***n***C**, where *n* is the number identifying the board. In this release, you can define a C interface for only one Policy Accelerator, nbhwpe0.

A usage demonstration application is included in the following location:

SDKinstallpath/demo/ODXFilter

For more information on binding and system ACE names, see *Developing Applications Using the IX-API SDK*and the *IX-API SDK Reference*.

New NCL Constructs

This release supports two new constructs in Network Classification Language (NCL):

- You can now specify conditional execution of classification rules. The new keyword **with** introduces a conditional clause, using the following syntax:

```
with boolean_exp {
   predicate pred_name { boolean_exp }
   rule rule_name { predicate } { action_ref }}
```

- You can now extend previously defined protocols by providing additional declarations for new fields outside the protocol definitions. Use the following syntax at the top level of the NCL file to add a field to an existing protocol definition:

  protocol_name.*field_name* **{***definition***}**

For more information on these features, see Chapter 7, "Network Classification Language," in the *IX-API SDK Reference.*

Diagnostic Utility

A new diagnostic utility, `pa100diag`, is available as part of the IX-API SDK installation at the following location:

SDKinstallpath/diagnostics/pa100diag

This tool allows you to test the installation of your Policy Accelerator 100. For more information, see Chapter 8, "Command-Line Tools," in the *IX-API SDK Reference*, and Chapter 2, "Troubleshooting the Policy Accelerator," in *Installing the IX-API SDK.*

New Demos

There are several new sample applications provided in the directory *SDKinstallpath*/demos. Both the new and existing sample applications are now operating-system–independent, and all applications now include Microsoft Visual C++ 6.0 project files (`.dsp` and `.dsw` files).

The following table briefly describes the techniques and concepts demonstrated by each of the sample applications.

Sample Application	Description
BasicApp	Shows how to construct the basic components of any application. Creates a single ACE that counts packets and uses an upcall to report the number of packets observed.
CountApp	Provides the source code used in the tutorial in *Developing Applications Using the IX-API SDK*. Simply counts packets and demonstrates the basics of creating applications.
Crosscall	Shows how to communicate between ACEs using crosscalls.
FilterApp	Shows how to construct a simple MAC address packet filter. An extension of BasicApp, in which only packets with a specific MAC destination address are forwarded. The ACE accepts packets from one MAC interface and forwards those that pass the filter criteria to the other MAC interface.

Sample Application	Description
ODXFilter	Shows how to transfer packets to and from a NIC whose driver has been customized using the Optimal Data Exchange (ODX) Protocol. Extends FilterApp to filter packets according to protocol before sending them to the NIC or dropping them.
FilterNic	Shows how to emulate a network interface card (NIC) in software. Each of the two ACEs accepts packets from one of the MAC interfaces, filters them, and sends those that pass the filter criteria to the host.
EventAppl	Shows how to use the `Event` class to schedule an activity that is not associated with data sets. This technique is far more efficient than using set element expiration for scheduling.
Firewall	Shows how to construct a real firewall application that determines which packets should be allowed into and out of the local area network, based on criteria that a user can set.
IPPairs	Shows how to use sets and searches to maintain a database about packet traffic. The application maintains a set of source–destination address pairs for IP packets that it has seen recently, and counts additional packets that match any known pair.
Killer	Shows how to construct an application that starts and stops the Resolver automatically. Only one such application can run on a Policy Accelerator at one time.
LoopApp	Shows how to construct a test to ensure that the IX-API SDK operating environment is configured correctly.
Simple	Shows how to use rules to classify packets of different types or with specific source or destination addresses.
StringSearch	Shows how to search for a string in packet data.
Tap	Shows how to construct a tap, in which the ACE “sniffs” a connection, looking at packets as they go by without redirecting them. In this case, the ACE keeps statistics about the packet flow.
TwoAceApp	Shows how to create and use an application with multiple ACEs. Extends BasicApp by making two separate ACEs in the same application that keep different counts.

Changes in This Release

In this release, the following features have changed or been eliminated:

Plumber and Factories Are No Longer Supported

The Plumber management application is no longer supported. Factories, which supported the Plumber, have been removed. The following changes in the host API remove Plumber-related complexities:

- The classes `NBFactory` and `NBPlumber` are no longer supported.
- In the `AceGroup` class constructor, you must pass `NULL` for the `argNBFactory` argument. Do not use the optional fourth argument, `list`.
- The `AceGroup` methods `getFactory` and `getFactoryID` have been removed.
- The `AceManager` method `export` has been removed.
- In the `NBAppl` constructor, the optional second and third arguments, `workingDirectory` and `cmdLine`, are no longer used.
- The Plumber-related methods of the base class `NBObject` have been removed. Only `getID` and `getType` remain.
- The `TargetManager` method `export` has been removed.

The sample applications in the `demo` directory have been updated to remove factories and related definitions.

NOTE: In this release, to ease porting, stubs remain for all of the removed functions and arguments, and existing code should run. In future releases, code that contains these references will break.

Microsoft Visual C++ 6.0 Project Is Required

The process for building applications on Windows NT has changed. In this release, you specify build and compilation information using the Microsoft Visual C++ 6.0 Project Setting dialog box, rather than in an exported makefile to be run on the command line. This enables you to build applications easily and interactively using Microsoft Visual C++ 6.0.

For details, see “Porting Applications from Previous Releases” on page 9.

System ACE NBSystemApp Alias Is No Longer Supported

When specifying system ACE names for bindings, the use of `NBSystemAppl` as an alias for the Policy Accelerator is no longer supported. Change all references to this alias to `nbhwpe0`. If an application attempts to create a binding using the alias, execution immediately terminates.

Integer Type Casting Is No Longer Automatic

When you pass an integer of type `nuint16` where an integer of type `nuint32` is expected, previous releases would cast the type automatically. This is no longer the case. In this release, you must explicity cast the type `nuint16` to the type `nuint32`. If you do not, the compiler reports an error.

Known Limitations of SDK 3.0

This section describes known problems or limitations in this IX-API SDK release.

General Limitations

Only Static Libraries Are Available

This release contains only the static versions of the IX-API system libraries.

Multiple Policy Accelerators Are Not Supported

This release does not fully support the use of multiple Policy Accelerators installed in a single computer.

Number of Concurrent ACEs

No more than four ACEs can run on a Policy Accelerator. If an application attempts to load more than four ACEs, the system stops.

MAC Interface Always in Promiscuous Mode

You cannot control the configuration of the promiscuous/nonpromiscuous mode of the MAC interface through the driver set-up dialog box. The MAC interface is always set to promiscuous. Check the Intel Web site for an example of how to provide equivalent filtering functionality with an application.

Stopping and Starting Applications

If you stop an application, Intel recommends that you stop and restart the Resolver before restarting the application or starting additional applications.

Stopping the Resolver

You must stop all running applications before stopping the Resolver. If you do not, unresolved references can cause unpredictable system behavior.

Stopping and Starting the PA100 Drivers

In accordance with the Windows NT NDIS driver standards, some parts of the driver software cannot be stopped and restarted while the system is operating. Therefore, to stop and restart the PA100 drivers, you must reboot the system.

NCL and ASL Libraries

NCL Code Size Limitations

NCL code size is currently limited to 1024 instructions. If the Resolver detects NCL object code that is too large after compilation, it rejects the load.

Set Restrictions

Set searches for longwords match only longword-aligned keys. For example, searches on ARP source/target protocol addresses will not match correctly, because the source address is not on a 32-bit packet boundary.

NCL Syntax Errors

When you run an application that has syntax errors in its NCL file, the Resolver displays the appropriate error messages and terminates.

Static Initialization Fails in Action Code

If you declare a static variable within an action code file, then attempt to initialize it statically to anything other than 0, the initialization is not executed and can result in a runtime error. To initialize a global variable to anything other than 0, perform the initialization as part of the `init_actions()` function. The use of static globals is not recommended.

Compiler Warnings

When compiling with Microsoft level 4 warnings, the following warnings can occur:

```
c:\ixsdk\include\nbapi\upcall.h(33) : warning C4121:
'UpcallHandler' : alignment of a member was sensitive to packing

c:\ixsdk\include\nbapi\nbfactory.h(27) : warning C4121:
'NBFactory' : alignment of a member was sensitive to packing
```

These warnings are not associated with any known problems.

Action Code Debugger (nbgdb)

Serial Port Connection Problem

Occasionally `getaceid` or `nbgdb` can fail to connect to the serial port. When this happens, the command (`getaceid`, or within `nbgdb`, `target remote /dev/tty00`) fails to return.

As a workaround, take the following steps:

1. Modify the file `/etc/remote`, changing the default baud rate for `tty00` to 57600.
2. Reset the port by connecting and disconnecting manually. For example, if you are using `tty00` (the first serial port), type the following at the shell prompt:

```
> tip tty00
> ~.
```

After making these changes, the commands connect properly, although the connection takes 80 seconds.

`nbgdb` Does Not Support More Than One ACE Sharing Action Code

If an application has more than one ACE with the same action code, the workaround is to duplicate the action code and change the global symbol names to associate with different ACEs.

Delete Breakpoints Before Detaching the Process

You must delete breakpoints before detaching the current process, or unexpected behavior will occur.

Use a Local Variable to Bypass Casting Bug in `gdb`

If a variable `varA` (a pointer) is cast to a class or more complex type (`typeT*`), `gdb` does not reflect its value correctly. As a workaround, define a local variable of the complex type (`typeT* varB`), and assign `varB = varA`. Use `varB` in `nbgdb` whenever you need to use `varA`.

ODX NIC Interface

Directly Connected NICs

In testing, Intel has found that when two Windows NT systems with Policy Accelerators and Intel® Pro/1000 NICs were connected directly together NIC-to-NIC, packets were not transferred reliably between the systems. When the systems are connected through a switch (as in normal practice) this problem does not occur.

Monitoring Features Not Supported for Interface C

This release does not support the interface property access functions (`NBInterfaceProp` class and `NBLinkwatch` class), nor does it support remote monitoring (`NBRmon` class), for interface C. (Interface C is available only if you have customized a NIC driver to communicate with the Policy Accelerator.)

Reported Incidents Fixed in This Release

- Incident #270: The NCL set size hint is no longer limited to 100 kilobytes (Kbyte). However, sets are not optimized for any size larger than 64 Kbyte.
- Incident #271: Constants are now computed at build time.
- Incident #584: Added `cecomp` option to specify include path.
- Incident #597: Resolver now boots the Policy Accelerator system software (avoiding `nb_init`).
- Incident #656: The `getaceid` command now works consistently, allowing the debugger to be used.

Porting Applications from Previous Releases

Porting from the SDK 2.0 Release

If you have applications working with SDK 2.0, they will work with the SDK 3.0 release on a Windows NT system with the following changes:

- The use of `NBSystemAppl` as an alias for `nbhwpe0` in bindings is no longer supported. Change all references to this alias to `nbhwpe0`.
- The Plumber and related API classes (such as `NBFactory`) are no longer supported. In this release, to ease porting, stubs remain for all of the removed functions and arguments, and existing code should run. In future releases, code that contains these references will break. You should remove all such references.
- The Windows NT-specific types `DWORD` and `PCHAR` are defined by the IX-API SDK system as the standard types unsigned long and `char *`, so existing code is OS-independent. However, it is recommended that you use the standard types in your applications.
- If your existing application is not already a Microsoft Visual C++ 6.0 project, you must create such a project for this release. You must specify build and compilation information in the project, rather than in an exported makefile.

To obtain the proper project settings, refer to the project files for the sample applications in `SDKinstallpath/demos`. Note the settings in the following sections of the Project Setting dialog box:

- C/C++ and Link tabs, Project Options
- Custom Build tab, Commands

The options and commands are slightly different for the release and debug versions of the applications. You must replace filenames and paths as appropriate for your application.

NOTE: To configure Microsoft Visual C++ 6.0 for use with the IX-API SDK, choose Options from the Tools menu, then, in the Options dialog box Directories tab, add the following pathname:

`SDKInstallPath/include`

You need to set this configuration only once. It is not necessary to do so for each project.

Documentation for This Release

This release includes new documents and updates to existing documents to improve completeness, provide more examples, and cover new features.

Installation Instructions

The installation instructions for the IX-API SDK software have been removed from the Release Notes and are now in their own document, *Installing the IX-API SDK*.

Documents Formats

Most documents are available in standard HTML, viewable from any recent browser; in PDF files, viewable using Adobe Acrobat Reader; and in HTML Help files for Internet Explorer.

New Documents

The following documents are new for this release:

- *Customizing a NIC Driver Using the ODX Protocol*

Documents Updated

All of the product documentation has been updated with expanded information and new material for the SDK 3.0 release:

- *Installing the IX-API SDK*
- *Installing a Policy Accelerator 100 Board*
- *Developing Applications Using the IX-API SDK*
- *IX-API SDK Reference*

Location

Documentation is located in `SDKInstallPath/documentation`. The file `index.html` contains links to all of the documents in all of the available formats.

Errata

In *Developing Applications Using the IX-API SDK*, the "Tutorial" chapter and the "Compiling Applications" chapter state that you can compile and link applications directly from the command line. In this release, you must use Microsoft Visual Studio. Similarly, the makefile information in the "Demonstration Application" appendix is out of date. The demos now include Visual Studio project files that you use to build the applications.

IX SDK Software Developer's Kit License Agreement

IMPORTANT: You (the "licensee") are consenting to be bound by this agreement if you do any of the following:

- **Click on the "accept" button**
- **Install or use the software**
- **Otherwise exercise any rights provided below to use the accompanying Intel™ IX API Software Developer's Kit (the "Intel SDK")**

Or, if applicable, you are bound by a currently effective written agreement regarding the use of the Intel SDK and signed by an authorized agent of you and by an officer of Intel.

If you do not agree to the terms of this agreement or such signed agreement, as applicable, then do not use or copy the Intel SDK, and contact the place from which you obtained it, if any of these terms are considered an offer, acceptance is expressly limited to these terms.

This Agreement sets forth the terms and conditions of your use of the accompanying Intel SDK, together with documentation provided to you by Intel. Any third party software that is provided with the Intel SDK with such third party's license agreement (in either electronic or printed form), and your use of such third party software, shall be governed by such third party's license agreement in addition to this Agreement. As used in this Agreement, Intel shall mean Intel Corporation, its affiliates, or its subsidiaries.

Users of the Intel SDK pursuant to this Agreement must either be individuals using the license on their own behalf or be employees or contractors of a corporation or other entity which has accepted the terms of this Agreement and on behalf of which the Intel SDK is being used, in which case the term "Licensee" in this Agreement refers to you and such entity.

1. Grant of License.

a. Subject to the terms of this Agreement, Intel grants to Licensee a worldwide, nonexclusive, nontransferable, nonassignable, nonsublicensable license (the "License") under Intel's copyrights to (i) copy the Intel SDK and associated documentation for internal use to integrate Applications for use with Intel Products, and (ii) to make and distribute as many copies of the integrated applications containing the Intel SDK as necessary. "Intel Products" means approved Intel Hardware listed in the datasheet provided with this Intel SDK. "Applications" means Licensee's current and future expected applications that will use the Intel SDK.

b. Accompanying the Intel SDK is specific source code ("Intel Source") such as ARM Compiler, ARM Debugger, Include Files, and reference applications that Licensee may incorporate into Applications during the integration process using the Intel SDK. Subject to the terms of this Agreement, Intel grants to Licensee a worldwide, nonexclusive copyright license to reproduce, distribute, and sublicense to third parties the Intel Source in Licensee's Applications for use with Intel Products. Licensee recognizes that when it uses the Intel SDK to create or compile Applications, a portion of the Intel SDK, the Intel Source, will be compiled and linked into or with the Applications.

2. Ownership of the Intel SDK. As between the parties, Intel retains title to and ownership of, and all proprietary rights with respect to, the Intel SDK, the Intel Source, and all copies and portions thereof, whether or not incorporated into or with other Software. The License does not constitute a sale of the Intel SDK, the Intel Source, or any portion or copy of it.

3. Restrictions; Licensee Obligations.

a. Any redistribution or duplication of any software, code, and or application derived from the Intel SDK shall require that the Intel InstallShield installation program be used for installation or Licensee agrees to incorporate the Intel file license.txt in its entirety into Licensee's install program. The Intel-provided file license.txt includes all relevant copyright notices, trademark notices, and any other notices. Except as specified in the applicable user documentation provided by Intel, Licensee shall not (and shall not allow any third party to) (i) decompile, disassemble, or otherwise reverse engineer or attempt to reconstruct or discover any source code or underlying ideas or algorithms of the Intel SDK by any means whatsoever, (ii) remove any product identification, copyright or other notices, (iii) retarget any Intel SDK to interoperate with products other than Intel Products, or (iv) provide, lease, lend, use for timesharing or service bureau purposes, or otherwise use or allow others to use the Intel SDK to or for the benefit of third parties.

Confidential information disclosed under this license agreement, including the existence and content of this Agreement, shall be considered "Confidential Information." Use and disclosure of such Confidential Information shall be governed by the terms of the Corporate Single Use Nondisclosure Agreement or other Nondisclosure Agreement, signed between the parties and incorporated into this Agreement by reference.

4. Termination of License for Cause. This agreement will remain in effect unless Intel terminates it due to a breach of its terms. Upon termination, Licensee will cease all use of the Intel SDK and promptly destroy or return to Intel all printed materials and copies of the Intel SDK and all portions thereof (whether or not modified or incorporated with or into other software) and so certify to Intel. Except for the License and except as otherwise expressly provided herein, the terms of this Agreement shall survive termination. Termination is not an exclusive remedy, and all other remedies will be available whether or not the License or the Agreement is terminated.

5. Limited Warranty and Disclaimer. **The Intel SDK is provided "as is" without warranty of any kind including, without limitation, any warranty of merchantability or fitness for a particular purpose or noninfringement. Further, Intel does not warrant, guarantee, or make any representations regarding the use, or the results of the use, of the Intel SDK or written materials in terms of correctness, accuracy, reliability, or otherwise.** Licensee understands that Intel is not responsible for and will have no liability for hardware, software, or other items or any services provided by any person or entity other than Intel.

6. Export Restrictions. Licensee agrees to fully comply with all applicable United States and EEC or other countries regulations and laws in effect now and hereinafter, including compliance with the U.S. Foreign Corrupt Practices Act and all export laws, restrictions, national security controls and regulations on the distribution or dissemination of Applications or Intel Products, technology, and information related to and/or exchanged under this Agreement. Licensee agrees not to export or reexport, or allow the export or reexport of the Intel SDK or any Intel Product, Intel Proprietary Information, or any direct product thereof in violation of any such restrictions, laws or regulations, or without all required licenses and proper authorizations, to Cuba, Libya, North Korea, Iran, Iraq, or Rwanda or to any Group D:1 or E:2 country (or national of such country) specified in the then current Supplement No. 1 to Part 740 of the U.S. Export Administration Regulations (or any successor supplement or regulations).

7. Government Contracts. The Intel SDK is provided with RESTRICTED RIGHTS. If Licensee is the Government or a Government contractor, use, duplication or disclosure by the Government is subject to the restrictions as set forth in subparagraph (c)(1)(ii) or the Rights in Technical Data and Computer Software Clause as DFARS 252.227-7013 and FAR 52.227-19, as applicable. Manufacturer is Intel Corporation, 1350 Villa Street, Mountain View, California 94041-1126.

8. Limitation of Remedies and Damages. **To the maximum extent allowed by law, Intel shall not be responsible or liable with respect to any subject matter of this agreement under any contract, negligence, strict liability, or other theory: (a) for loss or inaccuracy of data or cost of procurement of substitute goods, services or technology; (b) for any special, indirect, incidental, or consequential damages including, but not limited to, loss of profits; or (c) for any matter beyond its reasonable control.**

Distribution of the Intel SDK is also subject to the following limitations: Licensees (i) are solely responsible to your customers for any update or support obligation or other liability that may arise from the distribution, (ii) do not make any statement that your product is "certified," or that its performance is guaranteed, by Intel, (iii) do not use Intel's name or trademarks to market your product without written permission, (iv) shall prohibit disassembly and reverse engineering, and (v) shall indemnify, hold harmless, and defend Intel and its suppliers from and against any claims or lawsuits, including attorney's fees, that arise or result from your distribution of any product.

9. Transfer; Successors. Licensee shall not assign this agreement or any part of it except with Intel's prior written consent.

10. General. This Agreement shall be governed by and construed under the laws of the State of Delaware and the United States without regard to conflicts of laws provisions thereof and without regard to the United Nations Convention on Contracts for the International Sale of Goods. In any action or proceeding to enforce rights under this Agreement, the prevailing party shall be entitled to recover costs and attorneys' fees. If any provision of this Agreement is held by a court of competent jurisdiction to be illegal, invalid or unenforceable, that provision shall be limited or eliminated to the minimum extent necessary so that this Agreement shall otherwise remain in full force and effect and enforceable. No rights or licenses with respect to the Intel SDK or Intel Products are granted, other than those rights expressly and unambiguously granted in this Agreement. This Agreement constitutes the entire agreement between the parties relating to the subject matter hereof.

Copyrights and Trademark Notification

Intel: Copyright ©1998–2000 Intel Corporation. All Rights reserved. **Trademark**. Intel is a trademark of Intel Corporation.
*Other products and company names mentioned herein may be the trademarks of their respective owners.

UCB: Contains Software from The Regents of the University of California. Copyright ©1982, 1986, 1993, 1997-2000 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistribution of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

2. Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment: This product includes software developed by the University of California, Berkeley, Network Research Group at Lawrence Berkeley National Laboratory and its contributors.

4. Neither the name of the University nor the Laboratory nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the Regents and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose, are disclaimed. In no event shall the Regents or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

LCC: LCC Source Code from Addison Wesley Longman ("Licensor") from Christopher W. Fraser and David R. Hanson ("Authors"). LCC Source Code Copyright © 1995–2000 by David R. Hanson and AT&T. Reproduced by permission.

No warranty is made by Intel, the Licensor or the Authors of the LCC source code software, either express or implied, regarding the absence of defects in the LCC software, or its merchantability or fitness for a particular purpose. Intel, Licensor, and the Authors shall have no liability for damages of any nature arising out of any use, distribution, or modification of the LCC software, even if Intel, Licensor, or the Authors have been advised of the possibility of such damages.

GNU: Software coded using ARM Debugger and Compiler. Copyright ©1998–2000 Intel Corporation. All Rights reserved.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave., Cambridge, MA 02139, USA.

Intel Corporation
1350 Villa Street
Mountain View, CA 94041-1126
Tel: 650.567.9800
Fax: 650.567.9810
www.intel.com

Revision 4.0, June 2000